LAMK Lahden ammattikorkeakoulu
Lahti University of Applied Sciences

LOKITIEDON VISUALISOINTI

LAHDEN
AMMATTIKORKEAKOULU
Tekniikan ala
Digitaaliset teknologiat
Opinnäytetyö (ylempi AMK)
Kevät 2018
Pekka Tiirikainen

Lahden ammattikorkeakoulu
Digitaaliset teknologiat, insinööri (ylempi AMK)

TIIRIKAINEN, PEKKA Lokitiedon visualisointi

72 sivua

Kevät 2018

TIIVISTELMÄ

Tässä opinnäytetyössä toteutetaan lokitiedon visualisointijärjestelmä, joka piirtää ennalta määrätyt kuvaajat reaaliaikaisesti, kun lokitietoa syntyy. Visualisoinnin tehtävänä on nopeuttaa lokien tulkintaa ja parantaa lokitiedon ymmärtämistä.

Työssä valituksi tulivat Elasticin tarjoamat neljä sovellusta: Filebeat, Logstash, Elasticsearch ja Kibana. Filebeat lähettää lokitiedon Logstashille käsiteltäväksi. Logstash suodattaa ylimääräisen datan pois ja muokkaa jäljelle jäävän lokitiedon visualisoinnin vaatimaan muotoon. Logstash tallentaa muokatun lokitiedon Elasticsearchiin, josta Kibana lukee lokitiedon Web-käyttöliittymään. Kibanassa on useita kuvaajia ja useita raportointinäkymiä halutun tiedon esittämiseen.

Opinnäytetyö jakautuu alun teoriaosuuteen ja lopun käytännön toteutuksen kuvaukseen. Teoriaosuudessa käsitellään työn toteutuksen kannalta tärkeimmät teoriat: Big datan, visualisoinnin, lokitiedon ja radiotekniikan osalta. Lisäksi teoriaosuudessa esitellään valitut sovellukset ja niiden toiminta.

Tämän opinnäytetyön tuloksena toteutettiin työkalu langattoman viestijärjestelmän kehitysvaiheeseen radioarvojen vertailun helpottamiseksi. Työkalu mahdollistaa myös muidenkin lokitiedoissa olevien numeeristen tietojen visualisoinnin.

Avainsanat: visualisointi, lokitieto, suodatus, big data, elastic

Lahti University of Applied Sciences
Master's Degree Programme in Digital Technologies

TIIRIKAINEN, PEKKA: Visualization of log data

72 pages

Spring 2018

ABSTRACT

The objective of this thesis was to implement a system for visualizing log data in real time. The system draws predetermined graphs from collected log data. The purpose of visualization is to make it faster to interpret and understand log data.

Four applications by Elastic were chosen for the implementation: Filebeat, Logstash, Elasticsearch and Kibana. Filebeat sends log data to Logstash for processing. Logstash filters out extra information and modifies log data into a form that visualization requires. Logstash saves log data in Elasticsearch. Kibana reads log data from Elasticsearch to the web pages. There are many charts and dashboards in Kibana to present the desired log data.

The thesis is divided into the theoretical part and the description of the practical implementation. The theoretical part deals with the most important theories regarding the implementation of the work: big data, visualization, logging and radio technology. In addition, the theory section presents the selected applications and their function.

As a result of the thesis, a tool was created for the development stage of a wireless messaging system to facilitate comparison of radio values. The tool also allows other visualization of numerical data.

Key words: visualization, logging, filtering, big data, Elastic

SISÄLLYS

1 JOHDANTO 1

1.1 Tutkimuksen tausta 2

1.2 Tutkimuksen tarkoitus, tavoitteet ja tutkimuskysymykset 2

1.3 Tutkimusmenetelmä 3

1.4 Työn rakenne 4

2 BIG DATA JA VISUALISOINTI 5

2.1 Big datan määritelmä 5

2.2 Datasta tietämykseksi 6

2.3 Big datan hyödyntäminen 7

2.4 Big datan tehokas suodatus 8

2.5 Datan visualisointi 10

2.5.1 Datan visualisoinnin prosessi 10

2.5.2 Hyvä visualisointi 11

2.5.3 Kuvaajien visualisointi 12

2.5.4 Visualisoinnissa vältettäviä tapoja 14

2.6 Pylväsdiagrammi 14

2.7 Pohdinta Big datan ja datan visualisoinnin prosesseista 15

3 LOKIT JA NIIDEN KÄYTTÖ 16

3.1 Yleistä lokitiedostoista 16

3.2 Lokien käsittely 16

3.3 Lokien tyypit ja käyttö 17

3.4 Lokien säilytys ja suojaus 18

3.5 Lokitietoa koskeva lainsäädäntö 19

3.6 Lokien hyödyntäminen 20

3.7 JSON-tiedosto 20

4 RADIOTEKNIIKKA 22

4.1 Yleistä radiotekniikasta 22

4.2 Radioarvot 22

4.2.1 RSSI ja CINR 23

4.2.2 Modulaatio 24

4.3 Taajuuden merkitys radioyhteyden laatuun 24

5 ELASTIC-SOVELLUKSET 26

5.1 Valintaan vaikuttavat tekijät 26
5.2 Elastic-sovelluspaketti 26
5.3 Filebeat 27
5.3.1 Filebeat asetukset 28
5.3.2 Filebeat suodatus 29
5.3.3 Useammalle riville jakautuva lokitieto 29
5.4 Logstash 29
5.4.1 Logstash asetukset 30
5.4.2 Sisääntulo 31
5.4.3 Suodatin 31
5.4.4 Lähtö 32
5.4.5 Lisäosat ja koodekit 33
5.5 Elasticsearch 33
5.5.1 Klusteri (Cluster) ja solmu (Node) 34
5.5.2 Indeksi (Index) 34
5.5.3 Indeksin jakaminen paloihin ja replikointi 35
5.5.4 Elasticsearch mallien käyttö 35
5.6 Kibana 36
5.6.1 Kibanan käyttöliittymä 37
5.6.2 Discover 37
5.6.3 Visualize 38
5.6.4 Dashboard 38
5.6.5 Timelion 39
5.6.6 Dev Tools 39
5.6.7 Management 40
5.7 X-Pack 40
6 KÄYTÄNNÖN TOTEUTUS 42
6.1 Toteutuksen suunnitelma 42
6.2 Järjestelyjen yleiskuvaus 43
6.3 Elastic-sovellusten käyttöönotto 44
6.4 Filebeat käyttö 44
6.5 Logstash käyttö 46
6.5.1 Lokitiedoston sisältö 47
6.5.2 Logstash pipeline-asetukset 47
6.5.3 Logstash sisääntulo 48

6.5.4 Logstash suodatukset 48
6.5.5 Logstash suodatusasetusten rakentaminen 49
6.5.6 Ensimmäinen Logstash suodatin 50
6.5.7 Toinen Logstash suodatin 51
6.5.8 Kolmas Logstash suodatin 52
6.5.9 Logstash lähtö 53
6.6 Elasticsearch käyttö 54
6.7 Kibanan käyttö 55
6.7.1 Kibana indeksikuvio 56
6.7.2 Kibana Visualize 58
6.7.3 Kibana raportointinäkymä (Dashboard) 59
6.8 Kibanan raportointinäkymän monistaminen 60
6.9 Kibana raportointinäkymän päivittäminen 60
7 TUTKIMUSTULOKSET 63
8 JOHTOPÄÄTÖKSET JA JATKOKEHITYS 65
LÄHTEET 68

1 JOHDANTO

Lokitieto sisältää hyödyllistä tietoa järjestelmistä, niiden käytöstä ja käyttäjistä. Lokitieto ei kuitenkaan ole tekstimuotoisena suoraan hyödynnettävissä. Sen lukeminen ihmisen toimin on työlästä ja olennaista tietoa voi hävitä suureen tietomassaan. Visuaalisesta esityksestä ihminen pystyy luomaan nopeasti yleiskuvan ja ymmärtämään helpommin asioiden välisiä yhteyksiä kuin tekstimuotoisesta tiedosta.

Lokitietoa tuottavasta järjestelmästä riippuen lokitiedoissa saattaa olla paljon ylimääräistä dataa. Ennen visualisointia lokitieto on suodatettava ja muokattava visualisoinnin vaatimaan muotoon. Tästä syystä lokitieto on ensin tunnistettava ja sitä on opittava ymmärtämään, jotta sille pystytään tekemään tarvittava käsittely. Lokitiedon tunnistamisen jälkeen siitä pystytään suodattamaan ylimääräinen osa pois. Jäljelle jäävä, visualisoinnissa hyödynnettävä lokitieto on siivottava ja sen sisällön oikeellisuudesta on varmistuttava.

Laadukasta lokitietoa ja tilanteeseen sopivaa visualisointia hyödyntäen päästään kiinni olennaiseen tietoon ja pystytään ymmärtämään järjestelmien toimintaa ja mahdollisesti jopa tulevaisuutta paremmin. Opinnäytetyön tavoitteena on selvittää, kuinka tämä olennainen lokitieto löydetään ja saadaan esitettyä graafisessa muodossa lähes reaaliaikaisesti.

1.1 Tutkimuksen tausta

Langattoman viestijärjestelmän kehitysvaiheessa lokitiedot ovat hyvin keskeisessä osassa. Lokitietoihin tallentuu verkon ja laitteiden tilatiedot sekä virheilmoitukset. Yksi kehityksen osa-alue on käytännön testaus. Käytännön testauksen yksi osa-alue taas on radioyhteyden laadun tarkkailu ja vertaaminen aikaisempien sovellusversioiden vastaaviin arvoihin.

Käsin tehtävän työn vähentämiseksi on syytä etsiä tekniikasta apua lokitietojen tehokkaampaan hyödyntämiseen. Samalla myös pyritään reaaliaikaisempaan tietojen esittämiseen, kuin mihin ihmisen toimin on mahdollista päästä.

Osa lokitiedoista on radioverkon tilatietoa. Radioverkon tilatiedolla tarkoitetaan radioverkossa muodostuneiden yhteyksien laatua kuvaavia arvoja (radioarvot). Tällaisia yhteyksiä radioverkossa muodostuu kaikkien kuuluvuusalueella olevien laitteiden välillä. Kaikki radioarvot tallentuvat lokitietoon kahteen kertaan, koska arvot tallennetaan yhteysvälin kummankin radion kannalta katsottuna. Radioarvot muuttuvat ajan funktiona useasta eri syystä. Näitä syitä ovat esimerkiksi liike, yhteysväli, sää, käytetty sovellusversio sekä mahdolliset sovellusversion viat. Radioverkon lokitiedosta syntyy JSON-muotoinen tiedosto.

Työssä tiedoston sisällöstä tärkeimpiä tietoja ovat kullakin erillisellä yhteysvälillä modulaatio, RSSI (Received Signal Strength Indicator), CINR (Carrier to Interference Noise Ratio) ja tiedonsiirtonopeus. Lisäksi työn edetessä tarkastellaan, mitä muita arvoja on syytä ottaa mukaan kuvaajiin.

1.2 Tutkimuksen tarkoitus, tavoitteet ja tutkimuskysymykset

Työn tarkoituksena on etsiä käyttötarkoitukseen sopiva ohjelma, jolla voidaan lukea mahdollisimman yksinkertaisesti ja nopeasti JSON-tiedoston sisältö graafiseksi näkymäksi. Tiedostossa on paljon tietoa ja suurin osa siitä on tämän sovelluksen kannalta ylimääräistä. Sovelluksella

tulee saada suodatettua ylimääräinen tieto pois ja esitettyä vain tarpeellinen kiinnostava tieto käyttöliittymässä. Työssä täytyy rakentaa lokitiedon käsittelymalli, jolla ylimääräinen tieto saadaan suodatettua pois ja haluttu tieto saadaan näkyviin käyttöliittymään.

Tutkimuskysymykset ovat:

1. Millä sovellusohjelmalla JSON-muotoista lokitietoa pystytään esittämään graafisesti?
2. Kuinka tarvittava osa lokitiedosta saadaan esitettyä valitulla ohjelmalla graafisesti lähes reaaliajassa?
3. Kuinka datan suodatus kannattaa toteuttaa, kun kyseessä on suuri määrä lokitietoa ja vain murto-osa tarvitaan käyttöön?

1.3 Tutkimusmenetelmä

Tutkimusmenetelmäksi valittiin Design science eli suunnittelutieteellinen tutkimus. Tutkimusmenetelmän lähtökohtana on tieteen tutkimustulosten soveltaminen käytäntöön. Menetelmässä tutkimus jaetaan yleensä kahteen osaan: perus- ja soveltavaan tutkimukseen. Tutkimustyössä hyödynnetään perustutkimuksen tuloksia ja sovelletaan niitä kyseiseen käyttötarkoitukseen sopivaksi. Menetelmässä keskeistä on mallin rakentaminen ja sen toiminnan arviointi. Arvioinnin pohjalta mallia muokataan, kunnes sillä saadaan toteutettua haluttu tulos. (Järvinen & Järvinen, 103 -104.)

Menetelmä vaatii toimintaympäristön tuntemisen. Tavoitteena on parantaa nykyisten systeemien suorituskykyä. Yleistietämystä sovelletaan käsillä olevaan ongelmaan. Menetelmän lopputuloksena saadaan vastaus siihen kysymykseen, että miten alkutilanteesta päästään haluttuun lopputulokseen. (Järvinen & Järvinen, 103 -108.)

1.4 Työn rakenne

Valittu tutkimusmenetelmä näkyy työn rakenteessa siten, että alussa on teoriaosuus ja lopussa käytännön osuus. Teoriaosuudessa esitetään työn tekemisen kannalta keskeisimmät ja tarpeelliset teoriat. Käytännön osuudessa kuvataan valitut ratkaisut. Ratkaisut ovat syntyneet osin teorian pohjalta, mutta osin myös tutkimusmenetelmälle ominaisella rakentamisen ja arvioinnin menetelmällä. Työssä kuvataan lopputuloksena syntyvä ratkaisu.

Teoriaosuudessa aluksi toisessa kappaleessa käsitellään Big datan käsittelyn ja visualisoinnin teoria. Kolmannessa kappaleessa tutustutaan lokitietoa koskevaan teoriaan, lokitiedon keräämiseen ja säilyttämiseen liittyvään lainsäädäntöön. Kappaleessa neljä perehdytään radiotekniikan keskeisimpiin asioihin. Teoriaosuuden lopuksi kappaleessa viisi käsitellään Elasticin sovellukset, joilla varsinainen käytännönosuus toteutetaan.

2 BIG DATA JA VISUALISOINTI

2.1 Big datan määritelmä

Salon (2013, 10-17) mukaan Big data käsite viittaa kahteen asiaan, suureen datamäärän ja sen hyödyntämiseen. Dataa syntyy koko ajan valtavia määriä. Data voi olla esimerkiksi videota, ääntä, tekstiä ja kuvia. Sitä syntyy niin paljon, ettei kaikkea edes pystytä tallentamaan. Kuitenkin suuri datamäärä pitää sisällään vain osan tietoa, jota voidaan hyödyntää. Tästä syntyy haaste, kuinka oikea tieto löydetään ja kuinka se saadaan jalostettua hyödynnettävään muotoon.

Big datan hyödyntäminen korostuu yritysmaailmassa, jossa siitä hyödyntämällä pyritään saavuttamaan kilpailuetua muihin saman alan yrityksiin. Suuresta tietomäärästä on mahdollista löytää ne avainasiat, joihin yrityksen on panostettava menestyäkseen tulevaisuudessa. (Yuk & Diamond 2014, 29.)

Usein Big dataa määriteltäessä käytetään kolmen V-kirjaimen määritelmää: Volume (volyymi), Velocity (vauhti) ja Variety (vaihtelevuus). Volyymilla viitataan valtavaan datamäärään. Vauhdilla tarkoitetaan suurta nopeutta, jolla dataa syntyy ja sitä pitäisi pystyä hyödyntämään. Vaihtelevuudella tarkoitetaan datan lukuisia eri lähteitä ja muotoja. (Salo 2013, 21-22.) Myöhemmin määritelmään on lisätty vielä neljännellä V-kirjaimella alkava sana Veracity (totuudenmukaisuus). Totuudenmukaisuudella viitataan datalähteiden laatuun, onko käyttöön tuleva data oikeaa ja totuudenmukaista. Esimerkiksi sosiaalisessa mediassa jokainen voi kirjoittaa mitä haluaa, eikä sellaista tietolähdettä voida pitää välttämättä luotettavana. (Yuk & Diamond 2014, 32.)

Yksi tapa jakaa data eri tyyppeihin on jakaa se strukturoituun ja strukturoimattomaan. Strukturoidulla datalla on rakenne, eli sen tyyppi ja muoto on määritelty. Strukturoimattomalla datalla ei ole rakennetta, kuten esimerkiksi kuvat ja videot. Semistrukturoitu data on strukturoidun ja strukturoimattoman välimuoto. Semistrukturoitua dataa ovat esimerkiksi

valokuvat ja videot, joihin on liitetty metatietoa kuvaamaan kuvan tai videon sisältöä. (Dietrich et al. 2015, 6; Salo 2013, 21-22.)

2.2 Datasta tietämykseksi

Big dataan liittyen on tunnistettavissa neljä termiä (KUVIO 1): data (data), informaatio (information), tieto (knowledge) ja tietämys (wisdom). Data on raaka-aine, jota käsittelemällä on mahdollista löytää informaatiota. Informaatiota yhdistelemällä on mahdollista saada tietoa, josta on jalostettavissa tietämystä ja ymmärrystä. (Salo 2013, 26-27.)

KUVIO 1. DIKW-pyramidi (CSEstack 2017)

DIKW-pyramidi kuvaa kuinka suuri tietomäärä jalostuksen seurauksena supistuu pienemmäksi ja pienemmäksi. Samalla kun data määrä pienenee, oikealla käsittelyllä sen arvo kasvaa nousten aina tietämykseen asti. Tietämys on kuvaajan korkein taso ja samalla myös tavoitetila datan käsittelyssä. Datan jalostusta varten on kehitetty tekoälyä ja koneoppimista (machine learning), joiden avulla on mahdollista löytää

uutta informaatiota, joka on jalostettavissa tiedoksi ja tietämykseksi. (CSEstack 2017; Salo 2013, 26-27.)

2.3 Big datan hyödyntäminen

Ensimmäisiä Big datan kokeiluhankkeita on ollut lokianalyysit. Järjestelmät tuottavat paljon lokeja, joiden käsin analysointi on työlästä tai lähes mahdotonta. Lokeja tutkimalla on kuitenkin mahdollista ymmärtää järjestelmien ja laitteiden toiminnasta paljon. Pidemmällä aikavälillä on mahdollista jopa löytää piileviä vikoja ja ennalta ehkäistä vikatilanteita. Tässä on suuren tietomäärän oikea hyödyntäminen avain asemassa. (Salo 2013, 17-18.)

Big datan hyödyntäminen alkaa ongelman määrittämisestä. Ensiksi on selvitettävä, mikä on se ongelma, joka Big datan avulla halutaan ratkaista. Tällöin voi olla tarpeen luoda hypoteesi joidenkin asioiden välillä, mikä halutaan Big datan avulla todistaa oikeaksi tai vääräksi. Big datan hyödyntäminen vaati datan järjestelmällistä käsittelyä. Tällöin puhutaan Big datan prosessoinnista, mikä hyödyntää normaalin datan prosessoinnin vaiheitan. Datan prosessoinnin vaiheet (KUVIO 2) ovat datan *hankinta* (acquire)*, valmistelu* (prepare)*, analysointi* (analyze)*, raportointi* (report) *ja toiminta* (act). (Coursera 2017; Dietrich et al. 2015, 29-60.)

KUVIO 2. Big data prosessi (Coursera 2017)

Datan *hankintavaiheessa* selvitetään, mitä kaikkea dataa on käytettävissä kyseisen ongelman ratkaisemiseksi. Tällöin selvitetään kaikki mahdolliset lähteet, joilla voi olla merkitystä lopputulokseen pääsemisessä. (Coursera 2017; Dietrich et al. 2015, 29.)

Valmisteluvaiheessa data on tunnistettava ja siivottava. Tunnistamiseen kuuluu ymmärrys siitä, mitä data sisältää. Esimerkiksi, mitkä ovat sen raja-arvot. Eri lähteistä ladattu data on yhtenäistettävä samaan muotoon, jotta se on hyödynnettävissä. Datasta poistetaan tarpeettomat osat, eli data suodatetaan. (Coursera 2017; Dietrich et al. 2015, 29.)

Analysointivaiheessa esikäsiteltyä dataa hyödynnetään jollakin määritellyllä tavalla. Datan käsittelyyn suunnitellaan ja rakennetaan datan käsittelymalli. Mallin toiminta testataan. Tulosten perusteella tarkastetaan, toimiiko malli oikein ja hyväksytään malli käyttöön. Datan analyysiin kuuluu mm. luokittelu, klusterointi, regressio, graafinen analyysi ja assosiaatio analyysi. (Coursera 2017; Dietrich et al. 2015, 30.)

Luokittelussa data jaetaan ennalta määrättyihin ryhmiin. Klusteroinnissa saman tyyppiset tiedot jaetaan omiin ryhmiin. Klusteroinnissa ei tiedetä etukäteen, kuinka moneen ryhmään data jaetaan. Regressiossa ennustetaan numeerinen arvo. Graafisessa analyysissä hyödynnetään grafiikkaa yhteyksien löytämisessä eri kokonaisuuksien välillä. Assosiaatio analyysissä tiedon perusteella pyritään löytämään sääntöjä asiayhteyksien välille. (Coursera 2017; Dietrich et al. 2015, 30-43.)

Raportointivaiheessa keskeisimmät tulokset esitellään. Raportissa arvioidaan, mitä lisäarvoa analyysillä saadaan aikaiseksi ja vastaako tulokset tavoitetta. Raportointivaiheessa arvioidaan alussa asetettujen hypoteesien toteutuminen. (Coursera 2017; Dietrich et al. 2015, 30.)

Toiminnallisessa vaiheessa raportin tulokset julkaistaan. Tulosten perusteella mahdolliset muutokset siirretään käytäntöön. (Coursera 2017; Dietrich et al. 2015, 30.)

2.4 Big datan tehokas suodatus

Joissain tapauksissa Big datan suodattaminen voi olla haasteellista. Useista eri lähteistä ladattu datan voi olla hyvinkin monimuotoista. Kaikki data on muokattava samaan muotoon, jotta sitä voidaan hyödyntää

analysointivaiheessa. Dataan on tutustuttava ja se on ymmärrettävä ennen kuin sitä voidaan suodattaa siten, että vain tarpeellinen osa jää käyttöön. Joissain tapauksissa datan ymmärtäminen voi viedä pitkiäkin aikoja. (Frank 2012, 20.)

Suodatussääntöjen monimutkaisuus ja tietojen poistaminen tai säilyttäminen vaihtelevat datalähteistä ja ongelmasta riippuen. Strukturoidun datan suodattaminen ei vaadi paljoa työtä, mutta strukturoimattomassa datassa piileekin haaste. Yleensä se on ensin saatava määritettyä, standardisoitua ja ymmärrettyä, jotta sen jatkokäsittely on mahdollista. (Frank 2012, 21.)

Big datan standardisointia kehitetään siten, että se olisi helpompi jatkokäsitellä. Esimerkiksi on olemassa tapoja, joilla tunnistetaan hyviä ja huonoja sanoja tekstistä. Vie kuitenkin aikaa, että kehitettävistä standardeista muodostuu virallisesti hyväksyttyjä sääntöjä tai politiikkoja. (Frank 2012, 23.)

2.5 Datan visualisointi

Big datan myötä myös datan visualisoinnin tarve on lisääntynyt. Enää ei yksistään riitä, että esitellään taulukkomuotoista dataa. Data on visualisoitava sen paremman ymmärtämisen ja sitä kautta paremman datasisällön hyödyntämisen vuoksi. Visualisoidun kuvaajan tarkoitus on pakata esitettävä tieto niin, että se on nopeasti ymmärrettävissä. (Yuk & Diamond 2014, 1-43.)

Yksi tapa hyödyntää Big dataa on prosessoidun datan visualisointi. Tällöin analysoinnin tuloksena saatava tieto visualisoidaan sopivaan muotoon ja esitetään tuloksena. Tilanteesta riippuen visualisointia voidaan tarvita jo Big datan valmistelu- tai analysointivaiheissa. Valmisteluvaiheessa visualisointia voidaan tarvita datan ymmärtämiseksi. Analysointivaiheessa visualisointia taas hyödynnetään varsinaisten tulosten löytämisessä. (Dietrich et al. 2015, 36-50; Yuk & Diamond 2014, 41-64.)

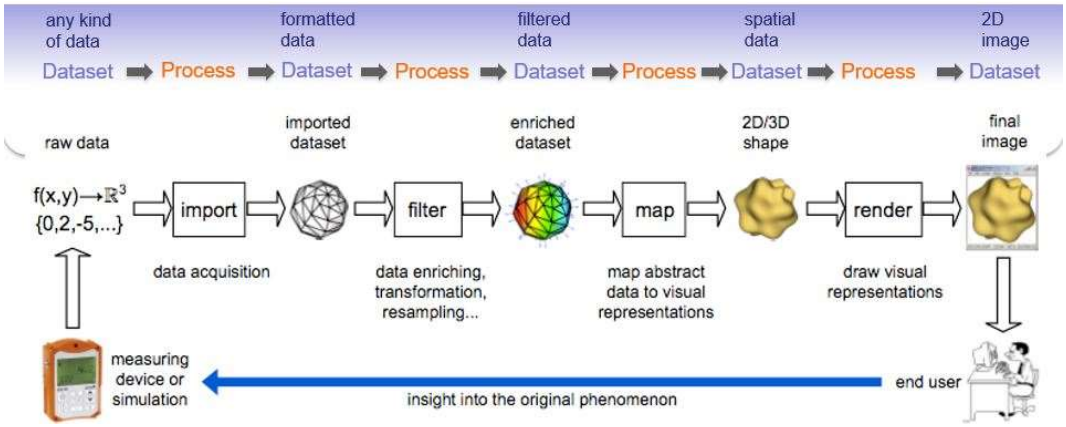
Datan merkityksen ymmärtäminen yritysmaailmassa on hyvin tärkeää. Tällöin on mahdollista ymmärtää paremmin menneitä, nykyisiä ja tulevia tapahtumia. Visualisoinnin tehtävä on helpottaa tätä haastetta edesauttamalla datan ymmärtämistä. Datan prosessointia hyödylliseksi tiedoksi visualisoinnin avulla kutsutaan nimellä Business Intelligence (BI). (Yuk & Diamond 2014, 8.)

2.5.1 Datan visualisoinnin prosessi

Datan visualisoinnin prosessin vaiheet (KUVIO 3) ovat: sisääntulo (import), suodatus (filter), liittäminen (map) ja kuvantaminen (render). *Sisääntulovaiheessa* määritetään, mitä ja missä sijaitsevaa dataan tarvitaan visualisoinnissa. *Suodatusvaiheessa* ylimääräinen data suodatetaan pois ja visualisoinnissa tarvittava data muokataan sopivaan muotoon. (Telea 2015, 123-125.)

Liittämisvaiheessa suodatetun tiedon kentät liitetään visualisoinnin vaatimaan rakenteeseen. Esimerkiksi tietystä kentästä piirtyy tietty viiva tai

pylväs kuvaajaan. *Kuvantamisvaiheessa* toteutetaan varsinainen raportointinäkymä kuvaajineen ja näyttöasetuksineen. (Telea 2015, 129-136.)



KUVIO 3. Datan visualisoinnin prosessi (Telea 2015)

Vaikka edellä kuvatussa mallissa datan visualisointi on jaettu neljään erilliseen vaiheeseen, niin tällaisia vaiheita todellisessa visualisoinnissa ei ole helposti tunnistettavissa. Tämän vaiheistuksen tehtävänä on antaa suuntaviivat visualisoinnille. Todellisuus riippuu käytetystä sovelluksesta ja itse visualisoitavastasta kohteesta. (Telea 2015, 145-146.)

2.5.2 Hyvä visualisointi

Kanervan (2017, 5) mukaan tiedon selkeä esittäminen on tärkeää paremman ymmärryksen ja toimintatapojen saavuttamiseksi. Tiedon jalostamisen taito on hyödyllinen, mutta sen oppimiseen on tehtävä jatkuvasti töitä.

Koponen et al. (2016, 32) toteavat, että visualisoinnin on oltava tarkoituksenmukainen. Esitystapa on valittava siten, että se tuottaa mahdollisimman selkeän lopputuloksen. Esitettävien visualisointien tulee olla oikein ja luotettavista lähteistä. Hyvä visualisointi mahdollistaa nopean

yleiskuvan aiheesta, mutta samalla se voi palkita lukijan, joka syventyy siihen paremmin.

Visualisoinnissa on erotettava tärkeä asia kohinasta. Kuvaaja on syytä pitää yksinkertaisina. Mitään ylimääräistä tietoa ei niihin pidä sijoittaa, vain olennainen tieto esitetään kuvaajassa. (Dietrich et al. 2015, 390.)

Tiedon visualisointi voidaan yleensä jakaa kahteen päätyyppiin: tutkiva ja selvittävä. Tutkivassa tiedon visualisoinnissa käyttäjä tutkii tietoa tarkemmin löytääkseen tiedosta uusia piirteitä. Selvittävässä visualisoinnissa esityksen on tarkoitus todistaa jokin väite. Yleisin virhe tiedonvisualisoinneissa on sotkea nämä kaksi visualisointityyppiä keskenään. (Kanerva 2017, 13.)

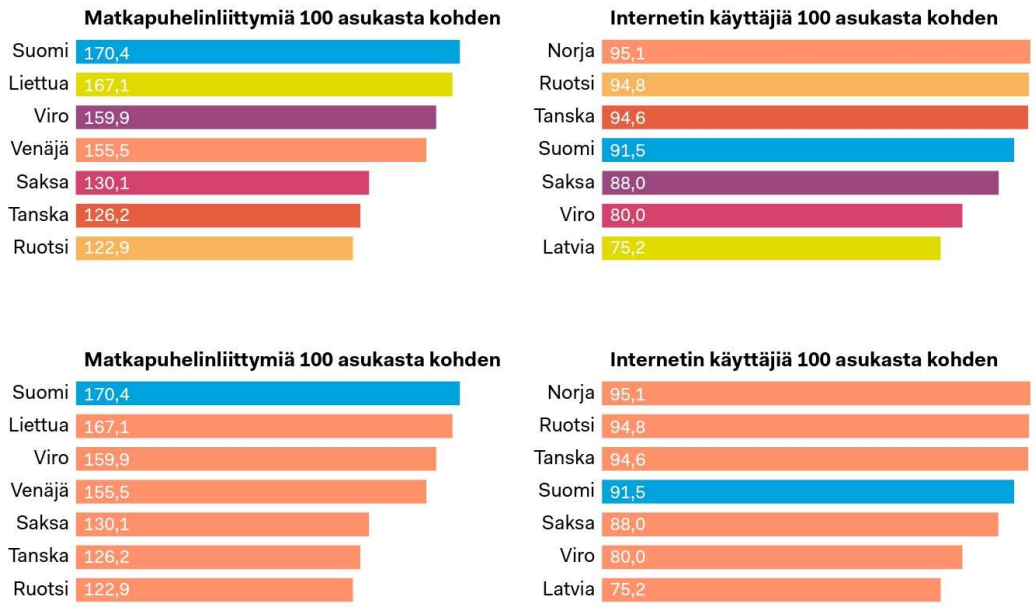
2.5.3 Kuvaajien visualisointi

Visualisoidun kuvaajan suunnittelussa keskeisin ratkaistava asia on suunnitella, mitä vertailuja visualisoinnilla halutaan mahdollistaa. Kuvaajan sanoma syntyy vertailusta. Vertailu mahdollistuu, kun verrattavia arvoja on useampia ja niiden välinen mittakaava on johdonmukainen. Suhteellisen visualisoinnin pääryhmiä ovat: lukumäärä tai suuruus, järjestys, kategoria, aika ja sijainti. Pääryhmistä poikkeava tieto on ennen visualisointia muutettava johonkin pääryhmistä. Muutoksessa voidaan käyttää apuna esimerkiksi luokittelua tai pisteytystä. (Koponen et al. 2016, 25-42.)

Samassa esityksessä olevien kuvaajien on syytä käyttää yhtenäisen tuntuista mallia ja niiden on oltava johdonmukaisia. Näin käyttäjän on helpompi vertailla tietoja ja ymmärtää asiasisältö, kun kuvaajien välillä rakenne on saman tyyppinen. Esitystavassa ei saa olla muutosta, vaan muutos on esitysaineistossa. (Kanerva 2017, 14; Koponen et al. 2016, 42.)

Värien käytössä on noudatettava myös johdonmukaisuutta. Samaa asiaa eri kuvaajissa esitettäessä on käytettävä samaa väriä. Varsinkin, jos kuvaajat ovat samalla sivulla. Mikäli tämä ei ole mahdollista, on syytä

harkita värien käyttöä siten, että korostetaan vain olennaista tietoa eri värillä ja muut tiedot käyttävät yhtä väriä (KUVIO 4). (Koponen et al. 2016, 43.)



KUVIO 4. Värien käyttö (Koponen et al. 2016, 42)

Kuvion 4 yläosassa olevista kuvaajista voidaan havaita sekavuutta värien käytössä. Esimerkiksi vasemmalla ylhäällä olevassa kuvaajassa Liettuaa kuvaa keltainen väri ja oikealla ylhäällä olevassa kuvaajassa keltainen onkin Latvian väri. Alemmissa kuvaajissa suomi on erotettu muista maista sinisellä värillä ja kaikki muut on kuvattu oranssilla. Näin alemmissa kuvaajissa on saatu selkeytettyä kuvaajien sanomaan.

Tiedon ymmärtämisen kannalta vertailtavien kuvaajien mittakaava on myös oltava sama. Mikäli mittakaava tai asteikko vaihtelee vertailtavilla kuvaajilla, on suuri riski, että käyttäjälle jää virheellinen käsitys kuvaajien välisistä suhteista. Näin ollen samalla sivulla esitettävien kuvaajien mittakaavan tulisi olla sama. Mittakaava on suositeltavaa pitää samana läpi koko esityksen. (Koponen et al. 2016, 44.)

2.5.4 Visualisoinnissa vältettäviä tapoja

Visualisoinnissa välteltäviä tapoja ovat 3D-kuviot, varjot ja muut kolmiulotteiset efektit. Ne ovat hienon näköisiä, mutta vaikeuttavat olennaisen tiedon löytämistä. Piirakkadiagrammin käyttämistä on myös vältettävä, jos esitettäviä asioita on enemmän kuin kolme. Piirakkadiagrammeja ei koskaan pitäisi laittaa vierekkäin vertailtavaksi, koska niiden vertaileminen on käyttäjälle hankalaa. (Kanerva 2017, 12.) Piirakkadiagrammi havainnollistaa huonosti osuuksien välisiä eroja. Jos verrattavia piirakkadiagrammeja on useampia, vertaileminen hankaloituu entisestään.

Tiedon visualisoinnissa tulee välttää sitä, että käyttäjän katse joutuu hankemaan tietoa pitkään visualisoinnista. Käyttäjän muistin varaan ei myöskään pidä jättää diagrammin osien otsikoita. Esimerkiksi usean värikoodin muistaminen piilottaa helposti olennaisen tiedon. (Kanerva 2017, 12.)

2.6 Pylväsdiagrammi

Tietoja vertailtaessa pylväsdiagrammi tuottaa parhaiten vertailuun sopivan kuvaajan. Pylväsdiagrammissa tieto voidaan esittää vakaa- tai pystysuunnassa riippuen esitettävän tiedon sisällöstä. Pystypylväsdiagrammissa (KUVIO 5) on yleensä molemmilla akseleilla numerotietoa kuvaava asteikko. Pystypylväsdiagrammi mahdollistaa ajan sijoittamisen X-akselille. Vaakapylväsdiagrammiin on helppo liittää pitkiä otsikkonimiä. Allekkain liitetty tieto mahdollistaa myös luettelomaisen esittämisen. (Koponen et al. 2016, 186.)

Pylväsdiagrammissa kuvaajalla on oltava selkeä otsikko ja tieto siitä, mitä X- ja Y-akselit esittävät. Lisäksi pylväille on määritettävä selkeät otsikot ja värit. (Yuk & Diamond 2014, 44.)

KUVIO 5. Esimerkki pystypylväsdiagrammista (Yuk & Diamond 2014, 44)

2.7 Pohdinta Big datan ja datan visualisoinnin prosesseista

Visualisoinnin ja Big datan lähteisiin perustuen voidaan niiden tavoitteita pitää varsin yhtenevinä. Kummassakin tavoitteena on lisätä tietämystä, ymmärrystä ja viisautta käsiteltävästä datasta. Kumpikin vaatii datan jalostamista siten, että olennainen osa saadaan käyttäjän hyödynnettävissä olevaan muotoon.

Datan visualisoinnissa ja Big datan käsittelyssä on huomattavan paljon samoja piirteitä. Visualisoinnin prosessi kylläkin painottuu enemmän tiedon esittämiseen, kun taas Big datan prosessi painottuu oikean tiedon löytämiseen ja analysointiin. Molempien prosessien sopiva yhdistäminen tuottaa oikein käsitellystä datasta hyvän ja hyödyllisen esityksen.

3 LOKIT JA NIIDEN KÄYTTÖ

3.1 Yleistä lokitiedostoista

Lokien tehtävänä on tallentaa järjestelmän tai sovelluksen ajonaikaiset tapahtumat. Lokeihin tallentuu tieto, miten järjestelmä tai sovellus on toiminut eri tilanteissa ja millaisia vikatilanteita on syntynyt. Loki on yleensä ylläpidon työkalu, mutta muitakin käyttötarkoituksia on olemassa. (Kyberturvallisuus 2016.)

Loki on käyttökelpoinen vain, jos siinä on kaikki olennaiset tiedot tallennettu. Käyttökelpoinen lokitieto vaatii vähintään seuraavat tiedot: aikaleima, tapahtuma, toimija, käyttöoikeus, tapahtuman lähde, kohde ja tila. Olennaista lokitietojen keräämisessä on, kuinka usein niitä tallennetaan. Lokitiedon tallennustahti riippuu kohdejärjestelmästä ja siihen ei ole olemassa yhtä selvää aikaväliä. Sopiva lokien tallennusväli löytyy vain kokeilemalla. (Kyberturvallisuus 2016.)

3.2 Lokien käsittely

Lokiohjeen (2009, 14) mukaan lokien käsittelyn määritelmä kattaa koko lokien elinkaaren. Elinkaari on suunniteltava käyttötarkoituksen ja arkistointivaatimusten mukaan. Elinkaari pitää sisällään lokien keräämisen, analysoinnin, säilyttämisen, luovuttamisen, arkistoinnin ja poistamisen.

Lokien käsittelyn suunnittelussa on huomioitava tarkoitussidonnaisuus, käyttötarve ja suunnitelmallisuus. Lokien käsittely on optimoitava niin, että ei tuhlata laitteisto- tai henkilöstöresursseja. Lokien käsittelystä aiheutuu aina kustannuksia, mutta se on välttämätöntä järjestelmän toiminnan, tietoturvallisuuden ja käyttäjien tietosuojan varmistamiseksi. (Lokiohje 2009, 19.)

Lokien käsittelyn neljä tärkeintä perusperiaatetta ovat:

- Käsittely perustuu määritettyyn tarpeeseen.

- Käsittely tapahtuu määritettyjen järjestelmien ja toimintatapojen mukaisesti.
- Lokien analysoinnin tulosten perusteella tehtävät toimet ovat ennalta määrätty.
- Järjestelmän sisältöön, käyttöön ja ylläpitoon liittyvien henkilöiden tietosuoja ja oikeusturva on huomioitu lokien käsittelyssä. (Lokiohje 2009, 19.)

Lokien käsittelyssä ja tallennuksessa on ehdottoman tärkeää huolehtia tietoturvasta. Lokien käyttöoikeudet ovat vain niille kuuluvilla henkilöillä. On hyvinkin mahdollista, että myös lokien käytöstä joudutaan muodostamaan lokitieto. (Kyberturvallisuus 2016.)

3.3 Lokien tyypit ja käyttö

Lokitiedostojen tulkitseminen edellyttää hyvää kyseisen järjestelmän tuntemusta. Lokitiedot voidaan jakaa ja luokitella usealla eri tavalla. Yksi vaihtoehto on jakaa ne ylläpito-, käyttö-, muutos- ja virhelokeihin. Usein kuitenkin lokien sisältö muodostuu edellä kuvattujen vaihtoehtojen yhdistelmistä. (Lokiohje 2009, 29-30.)

Lokitiedostot ovat ylläpidon tärkein työkalu. Ongelmien esiintyessä lokitiedoista löytyy virheilmoitukset ja mahdolliset vian aiheuttajat. Myös mahdollisista järjestelmään murtautumisista tai murtautumisyrityksistä jää lokiin merkintä ja näin havaitaan syntynyt ongelmatilanne. (Kyberturvallisuus 2016.)

Uusien järjestelmien kehitysvaiheessa lokit ovat tärkeitä järjestelmän toiminnan varmistamiseksi sekä mahdollisten vikojen löytämiseksi. Kehitysvaiheessa muodostettavalta lokitiedolta vaaditaan hieman eri asioita kuin järjestelmän siirryttyä tuotantokäyttöön. Tämä on syytä huomioida järjestelmän toteutuksessa siten, että lokitiedostoon kirjattavia tietoja voidaan tarpeen ja käyttötarkoituksen mukaan säätää.

3.4 Lokien säilytys ja suojaus

Lokit sisältävät usein järjestelmiin, tietototurvaan ja henkilöihin liittyvää tietoa, siksi lokit on suojattava. Lokien luottamuksellisuus ja eheys on turvattava. Myös lokien varmuuskopioinnista ja vanhojen lokien poistosta on huolehdittava. Lokien saatavuuteen on kiinnitettävä huomiota siten, että ne ovat saatavilla kohtuullisen helposti. Mikäli lokien sisältämään tietoon on hankala päästä käsiksi, lokit voivat jäädä tutkimatta ja esimerkiksi mahdolliset tietomurrot tai vikatilanteet jäävät havaitsematta. (Lokiohje 2009, 57; Kyberturvallisuus 2016.)

Lokiohjeessa (2009, 58) todetaan, että lokikierrolla tarkoitettaan luotavan lokitiedoston sulkemista ja uuden avaamista. Lokikierto voidaan aikauttaa tietyn aikataulun, koon tai rivimäärän mukaan. Aikataululla tarkoitetaan sitä, että esimerkiksi kerran vuorokaudessa avataan uusi tiedosto, johon seuraavan vuorokauden lokit kirjoitetaan. Koko määrityksessä taas uusi tiedosto avataan, kun ennalta määrätty tiedostokoko saavutetaan. Yksi vaihtoehto on myös määrittää rivimääräehto, jonka jälkeen avataan uusi tiedosto. Tiedoston koko- ja aikamäärityksillä pyritään siihen, että lokitiedon koko ei kasva liian suureksi ja sen käytettävyys on mahdollista.

Lokitietoa tuottavien järjestelmien kellonajan synkronoinnista on varmistuttava esimerkiksi aikapalvelun (NTP, Network Time Protocol) avulla. Lokitiedoissa oleva oikea aikaleima on ehdoton edellytys lokien käytettävyydelle. Muutoin tapahtumien selvittäminen jälkikäteen on hankalaa tai jopa mahdotonta. (Lokiohje 2009, 61.)

Kaikissa tilanteissa lokit on suojattava siten, että ne eivät altistu tahattomalle tai tahalliselle muokkaamiselle ja tuhoamiselle. Esimerkiksi haittaohjelmat saattavat poistaa tai muokata lokeja niiden tarkoituksiin sopiviksi. Lokien suojauksen pääperiaate on, että järjestelmän muodostamia lokeja ei pystytä jälkikäteen muokkaamaan. Lokit on ainoastaan mahdollista tuhota niiden säilytysajan päättyessä. (Lokiohje 2009, 57-61.)

3.5 Lokitietoa koskeva lainsäädäntö

Henkilötietolaki määrittää myös lokienkäsittelyn vaatimuksia ja velvollisuuksia, kun lokitiedoissa on henkilölaissa määritettyjä tietoja. Tällaisia tietoja ovat esimerkiksi henkilönnimi ja IP-osoite. Laissa viitataan henkilörekisteriin. Lokitiedosta syntyy henkilörekisteri, kun siinä on henkilötietoja. Näin ollen henkilörekisteriä koskevat vaatimukset voivat koskea myös lokitietoja. (Lokiohje 2009, 71-72.)

Muita lokitiedon julkisuutta, säilytystä ja käsittelyä määrittäviä lakeja ovat:

- Laki viranomaisen toiminnan julkisuudesta (621/1999)
- Laki yksityisyyden suojasta työelämässä (759/2004)
- Sähköisen viestinnän tietosuojalaki (526/2004)
- Julkisuusasetus (1030/1999)
- Laki sananvapauden käyttämisestä joukkoviestinnässä (460/2003)
- Laki yhteiskuntapalvelujen tarjoamisesta (458/2002)
- Arkistolaki (831/1994) (Lokiohje 2009, 73-80.)

Edellä mainittujen kansallisten lakien lisäksi EU:n tietosuoja-asetus GDPR (General Data Protection Regulation) (2016/679) ottaa kantaa henkilötietojen käsittelyyn, luovutukseen ja liikkuvuuteen EU-alueella. GDPR-asetuksen tarkoitus on parantaa henkilörekisteriin kuuluvien oikeuksia valvoa henkilötietojen käsittelyä ja säilyttämistä sekä määrittää velvoitteet rekisterin ylläpitäjälle. Asetuksen nojalla henkilörekisteriin kuuluvilla on oikeus tarkistaa rekisteristä omat tietonsa ja oikeus saada oikaistua ne. Henkilörekisteriin kuuluvalla on myös oikeus määrittää henkilötietojen luovutusta ja käyttötarkoitusta koskevat seikat. GDPR tietosuoja-asetus koskee myös niitä EU:n ulkopuolisia yrityksiä, joiden rekisterissä on EU-kansalaisia. (GDPR haltuun, 2-3.)

3.6 Lokien hyödyntäminen

Lokiohjeen (2009, 47-48) mukaan lokitietojen analysointi on usein vaativin, mutta myös tärkein osa-alue. Laajoissa järjestelmissä analysointiin onkin syytä ottaa avuksi jokin lokien hallintaan ja analysointiin tarkoitettu työkalu. Lokimerkintöjen käsittelyssä on syytä hyödyntää suodatusta. Tunnettaessa, mitkä lokimerkinnät ovat tärkeitä ja mitkä eivät, voidaan lokien käsittelyyn rakentaa automatiikka. Automatiikka voidaan rakentaa esimerkiksi siten, että tuotetaan kaksi erillistä raporttia. Yksi lokimerkinnöille, jotka on tunnistettu tärkeiksi ja toinen lokimerkinnöille, joita ei vielä tunneta.

Epätavallisten ja haitallisten lokimerkintöjen tunnistamiseen on syytä rakentaa hälytysmekanismi, jotta syyn aiheuttajan jäljille päästään mahdollisimman nopeasti. Mikäli lokeja joudutaan analysoimaan manuaalisesti, on analysointi huomattavasti tehokkaampaa ja helpompaa, kun lokeista on suodatettu kaikki tavalliset tapahtumat pois. Tavallisista lokimerkinnöistä voi esimerkiksi olla tarpeellista raportoida vain määrät. (Lokiohje 2009, 47-48.)

3.7 JSON-tiedosto

JSON-tietotyyppi on yksi lokitiedon tiedostomuoto. JSON (JavaScript Object Notation) on avoimen standardin tiedon välitykseen tarkoitettu datan esitysmuoto. Se perustuu JavaScriptiin, mutta silti se on kuitenkin täysin kieliriippumaton tietomuoto. (JSON 2017; Wiki freepascal 2017.)

JSON-tietoa on ihmisen kohtuullisen helppo lukea ja kirjoittaa. Lisäksi tietokoneiden on helppo tuottaa ja muokata sitä. XML-tietomuotoon verrattuna JSON-tietomuoto on helpompi ihmisen ymmärtää. JSON-tiedoston sisältö muodostuu olioihin kuuluvista avain - arvo pareista. Arvo voi olla merkkijono, numero, taulukko, kokoelma, tosi, epätosi tai tyhjä. JSON-oliot voivat myös muodostua useista sisäkkäisistä olioista. Sisäkkäisyyden lisääntyessä tiedoston lukeminen ihmisen toimin muodostuu vaikeammaksi tai jopa mahdottomaksi. JSON-tiedostossa

käytetään päätettä ”.json”. JSON sääntöjä ja ohjeita on määritetty RFC 4627 dokumentissa. (JSON 2017; Wiki freepascal 2017.)

JSON-tiedoston rakenne voidaan muodostaa kahdella eri tavalla. Yksi olio voi olla yhdellä rivillä (KUVIO 6) tai jaoteltuna useammalle rivillä (KUVIO 7). Kummassakin tapauksessa muoto on muuten sama, mutta rivivaihdot selkeyttävät ihmisen lukemista ja havainnointia. (Wiki freepascal 2017.)

```
{"nimi1":arvo1, "nimi2":arvo2}
```

KUVIO 6. JSON-olio yhdellä rivillä (Wiki freepascal 2017)

```
{
        "nimi1": arvo1,
        "nimi2": arvo2
}
```

KUVIO 7. JSON-olio rivitettynä (Wiki freepascal 2017)

4 RADIOTEKNIIKKA

4.1 Yleistä radiotekniikasta

> *”Radiotekniikka on radioaaltojen noudattamien luonnonlakien tuntemiseen perustuvaa toimintaa, jonka avulla radioaaltojen tarjoamat mahdollisuudet saatetaan palvelemaan ihmisen päämääriä” (Lehto & Räisänen 2007, 11).*

Kuten Lehto ja Räisänen (2007) ovat kirjassaan kertoneet, radiotekniikassa pyritään hyödyntämään radioaaltojen etenemistä ilmassa mahdollisimman tehokkaasti. Yksi merkittävä tekijä radioaallon etenemiseen on kantoaalto ja sen taajuus. Kantoaalto on perustaajuus, johon siirrettävä tieto liitetään mukaan modulaation avulla. Modulaatiosta kerrotaan tarkemmin luvussa 4.2.2. (Kosola 2013, 100-101; Lehto & Räisänen 2007, 12.)

Vaatimuksena radioliikenteen onnistumiselle on taajuuksien määrittäminen eri järjestelmien käyttöön. Taajuusjaolla pyritään varmistamaan, että eri järjestelmille on riittävästi käyttökelpoisia ja häiriöttömiä radiotaajuuksia. Taajuuksien käytöstä on sovittu kansainvälisin sopimuksin. Kansainvälisesti taajuuksista vastaa International Telecommunication Union (ITU). (Lehto & Räisänen 2007, 12.) Suomessa taajuusjaosta vastaa Viestintävirasto. Viestintäviraston sivuilla on julkaistu Suomessa voimassa oleva taajuusjakotaulukko. Taajuusjakotaulukossa esitetään jaetut taajuudet ja niiden käyttökohteet. (Viestintävirasto 2017.)

4.2 Radioarvot

Kosolan (2013, 104-115) mukaan radioarvoilla kuvataan yhteyden laatua lähetys- ja vastaanottopisteiden välillä. Yhteyden laatuun vaikuttavia tekijöitä on paljon. Pelkkä lähetysteho ei ole ainoa asia, mikä vaikuttaa yhteyden laatuun. Muita yhteyden laatuun vaikuttavia tekijöitä ovat mm. laitteisto, sää, etäisyys, taajuus, antennit, antennikaapelointi, yhteysvälin lähiesteet ja maaston muodot. Antennikaapeloinnissa yhteyden laatuun

vaikuttavia tekijöitä ovat antennikaapelin pituus, laatu, suojaus, liitokset, asennetun kaapelin taivutussäde. Antennin määrä ja monitie-eteneminen vaikuttavat myös radioyhteyden laatuun.

Kaikki yhteyden laatuun vaikuttavat tekijät eivät kuitenkaan muodostu yhteysvälistä ja sen laitteista, vaan lisäksi on olemassa vielä ulkopuolisia tekijöitä, kuten esimerkiksi samalla taajuusalueella toimivat muut lähetteet. Vuoden ajalla ja auringon aktiivisuudella on myös vaikutus radioyhteyden laatuun. (Kosola 2013, 86-88.)

Radiotekniikka käsitteeseen kuuluu myös radiojärjestelmien tutkiminen, kehittäminen ja mittaaminen (Lehto & Räisänen 2007, 11). Tutkimus- ja kehitystyössä yhteysväliin vaikuttavat tekijät pyritään pitämään samoina kuin aikaisemmissa mittauksissa. Tämä on kuitenkin lähes mahdotonta maastossa testattaessa. Sää ja vuodenajat aiheuttavat muutoksia jatkuvasti. Lisäksi testipisteen paikka ihmisen määrittämänä on lähes mahdotonta saada juuri samaksi eri testien välillä. Ainoastaan laboratorio-oloissa on mahdollista saada täysin samat ympäristötekijät eri testien välille.

4.2.1 RSSI ja CINR

RSSI (Received Signal Strength Indicator) kertoo vastaanotetun tehon voimakkuuden. Sillä ei ole varsinaisesti yksikköä, mutta yleensä se ilmoitetaan desibeleinä suhteessa milliwattiin (dBm) tai milliwatteina (mW). RSSI-arvo ilmaistaan negatiivisessa muodossa. Mitä lähempänä arvo on nollaa, sitä vahvempi vastaanotettu signaali on. (Nuaymi 2007, 192.)

CINR (Carrier to Interference and Noise Ration) kuvaa radiosignaalin laatua. CINR lasketaan kantoaallon tehon, kohinatehon ja häiriöiden suhteesta. CINR-arvo ilmoitetaan yleensä desibeleinä (dB). (Nuaymi 2007, 193.)

4.2.2 Modulaatio

Lehto & Räisänen (2007, 225) kertovat, että modulaation tehtävänä on sovittaa siirrettävä tieto kantoaaltoon. Modulaatio kuvaa tiedonsiirtokykyä kyseisellä radioyhteydellä. Sopivan modulaation tehtävänä on hyödyntää käytettyä taajuusaluetta mahdollisimman hyvin, sietää kohinaa, häiriöitä ja häipymiä.

IEEE 802.16 standardi tukee neljää modulaatiota, jotka ovat BPSK, QPSK, 16-QAM ja 64-QAM. BPSK on datansiirtokyvyltään heikoin modulaatio, mutta häiriön sietokyvyltään se on paras. Vastaavasti 64-QAM modulaation on datansiirtokyvyltään paras, mutta sen häiriön sietokyky on heikoin. IEEE 802.16 standardi tukee adaptiivista modulaatio, jolloin modulaatio vaihtuu yhteyden laadun mukaan. Yhteyden laadun ollessa hyvä, käytetään korkean tason modulaatiota, kun taas yhteyden laadun ollessa huono, käytetään matalan tason modulaatiota. (Nuaymi 2007, 45-48.)

4.3 Taajuuden merkitys radioyhteyden laatuun

Radioaallot etenevät parhaiten näköyhteysreitillä (LOS, Line of Sight), jolloin yhteys välillä ei ole mitään esteitä välissä. Taajuuden kasvaessa vaatimus näköyhteydestä kasvaa, kun taas pienemmille taajuuksille näköyhteysvaatimus ei ole niin merkittävä. (Kosola 2013, 75-77.)

Näköyhteysvaatimusraja on UHF-taajuusalueesta alkaen (Kosola 2013, 76). UHF-taajuusalue on 300 MHz – 3000 MHz. Taulukossa 1 on esitetty radioaaltojen taajuusalueet ja niiden nimet. (Lehto & Räisänen 2007, 10.)

Kosolan (2013, 87-88) mukaan taajuus vaikuttaa myös radioaaltojen häiriöiden etenemiseen. Matalilla taajuuksilla häiriöt voivat edetä pitkiä matkoja. Taajuuden kasvaessa vain lähellä olevat häiriölähteet vaikuttavat.

TAULUKKO 1. Radioaaltojen taajuusalueet (Lehto & Räisänen 2007, 10)

Lyhenne	**Pitkänimi**	**Taajuusalue**
VLF	Very Low Frequency	3-30 kHz
LF	Low Frequency	30-300 kHz
MF	Medium Frequency	300-3000 kHz
HF	High Frequency	3-30 MHz
VHF	Very High Frequency	30-300 MHz
UHF	Ultra High Frequency	300-3000MHz
SHF	Super High Frequency	3-30 GHz
EHF	Extremely High Frequency	30-300 GHz

5 ELASTIC-SOVELLUKSET

5.1 Valintaan vaikuttavat tekijät

Työssä etsittiin sovellusta, joka mahdollistaa lokitiedon suodatuksen lisäksi myös sen esittämisen graafisesti. Valintapäätökseen vaikuttavia tekijöitä olivat ensisijaisesti tietenkin se, että sovelluksella saadaan toteutettua JSON-muotoisen lokitiedon suodatus ja visualisointi. Lisäksi valintaan vaikuttivat valitun sovelluksen skaalautuvuus, dokumentaatio, tieto vastaavista käyttökohteista ja sovellukselle saatava tuki.

Eri sovellusvaihtoehtoja on tarjolla lukuisia. Kuitenkin edellä kuvattuja vaatimuksia peilaten yksi sovelluspaketti tuli valituksi. Elasticin valmiista sovelluksista löytyy kattava dokumentaatio, videoita ja käytännön esimerkkejä eri keskustelupalstoilta. Elastic on myös julkaissut sovelluspaketin suurimpia käyttäjiä, joita ovat muiden muassa Dell, eBay, Netflix ja Symantec. Elasticin sovelluksilla on mahdollista vastaanottaa useita eri tyyppisiä lokitietoja. Sovellukset mahdollistavat myös ympäristön skaalautuvuuden käyttötarkoituksen vaatimusten mukaisesti yksittäisestä palvelimesta laajaan klusteroituun ympäristöön.

5.2 Elastic-sovelluspaketti

Työssä valituksi sovellukseksi valikoitui Elasticin valmistamat Elastic Stack sovellusperhe. Elastic Stack sovellusperhe muodostuu neljästä erillisestä ohjelmasta Logstash, Elasticsearch, Kibana ja Beats. Nämä neljä ohjelmaa yhdessä muodostavat keskitetyn lokien hallinnan ja reaaliaikaisen visualisoinnin. Logstash jäsentää ja Kibanalla tarkastellaan graafisesti kerättyjä lokeja. Elasticsearch indeksoi Logstashin jäsentämät lokit Kibanan ymmärtämään muotoon. (Elastic 2017.)

Beats ei ole varsinainen ohjelman nimi, vaan ohjelmien ryhmän nimi. Beats ohjelmia Elasticillä on tarjolla useita käyttötarkoituksesta ja käyttöjärjestelmästä riippuen. Beatsit ovat pieniä ohjelmia, joiden tehtävänä on lähettää lokitiedot alkuperäisestä lokeja tuottavasta

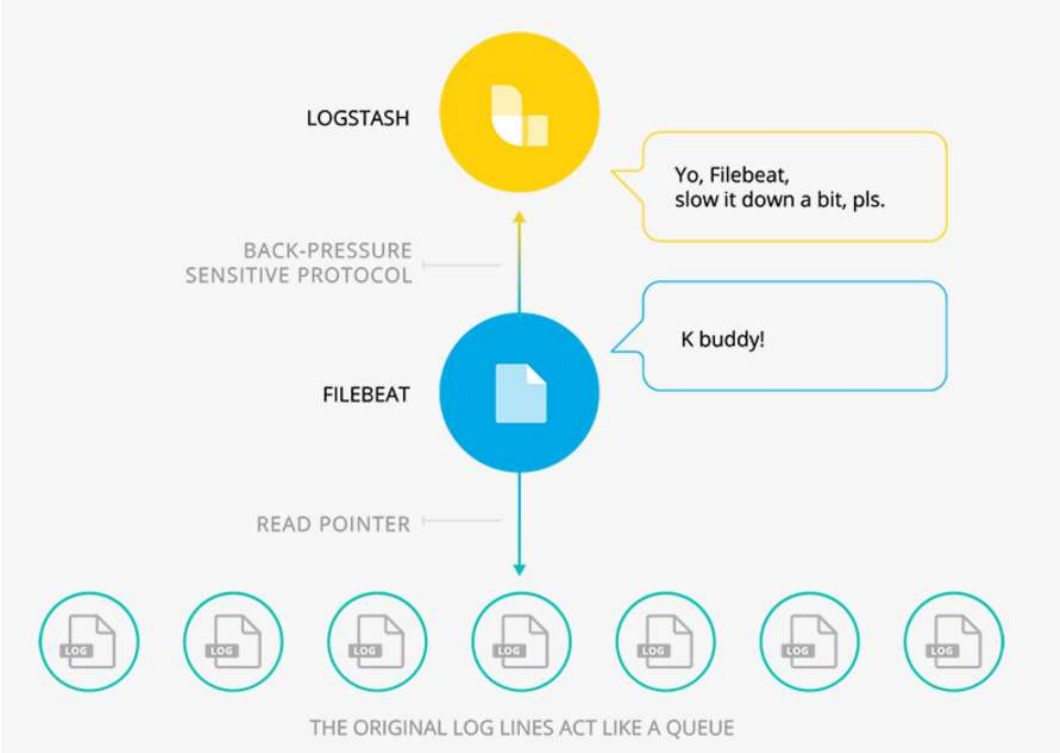
ympäristöstä Logstashille. Beat-sovellukseksi valittiin Filebeat, koska se on tehty lokitiedon välittämiseen Linux-käyttöjärjestelmälle. (Elastic 2017.)

Filebeat, Elasticsearch, Logstash ja Kibana ovat suosittuja ohjelmia. Ne ovat saman tahon ylläpitämiä ja kehittämiä, joten ne ovat suunniteltu toimimaan hyvin toistensa kanssa. Avoin lähdekoodi myös takaa sen, että minkä tahansa näistä kolmesta vaihtoehdosta voi aina korvata vaihtoehtoisella sovelluksella. Elastic Stack sovellukset on saatavissa Windows, Linux ja MAC käyttöjärjestelmille. (Elastic 2017.)

5.3 Filebeat

Filebeat on erillinen pieni ohjelma, jonka tehtävänä on lähettää lokitiedostot Logstashille. Filebeat-sovelluksella on mahdollista kerätä useasta eri lähteestä lokitiedostot yhteen ja lähettää ne Logstashin käsiteltäväksi. Filebeat on suunniteltu luotettavaksi ja nopeaksi tiedonvälityskomponentiksi. Beat sisääntulo suodattimen käyttö minimoi resurssivaatimukset Logstash-sovelluksessa. (Filebeat 2017.)

Filebeat-sovelluksen tehtävänä on myös huolehtia, että sen kohdesovellus, esimerkiksi Logstash, ei joudu ylikuormitustilanteeseen. Filebeat hidastaa lokitiedon lähettämistä, mikäli ylikuormitustilanne on syntymässä (KUVIO 8). Kun ylikuormitustilanne on ohi, Filebeat palaa takaisin alkuperäiseen lokintiedon lähetysnopeuteen. (Filebeat 2017.)



KUVIO 8. Filebeat toiminta (Filebeat 2017)

5.3.1 Filebeat asetukset

Filebeat-asetukset sijaitsevat asennuskansiossa filebeat.yml tiedostossa. Debian-käyttöjärjestelmässä asetustiedosto on oletuksena /etc/filebeat/filebeat.yml. Asetustiedostossa kerrotaan luettavan lokitiedon tyyppi, sijainti ja mihin lokitieto lähetetään. Lokitiedon sisääntulon tyyppi voi olla log tai stdin. Log-tyyppi lukee kaikki rivit ja stdin-tyyppi lukee vain erikseen määritetyt tiedot (input_type). Lokitiedostoja ja niiden sijainteja voidaan määrittää useita. Myös *-merkin käyttö on sallittua lokitiedoston sijaintia ja nimeä määritettäessä Filebeatin luettavaksi. (Filebeat Docs 2017a.)

Filebeat sovellus ylläpitää tietoa jo lähetetyistä lokitiedoista. Mikäli tulee tarve lähettää vanhat lokitiedot uudelleen, on historiatieto poistettava ja käynnistettävä Filebeat uudelleen. Tällainen tilanne voi syntyä esimerkiksi

ympäristön asennus- ja testausvaiheessa. Historiatieto sijaitsee tiedostossa /var/lib/filebeat/registry. (Filebeat Docs 2017b.)

5.3.2 Filebeat suodatus

FIlebeat-sovelluksella on mahdollista tehdä datan siivoaminen käyttötarkoitukseen sopivaksi. Tällaisia toiminteita ovat esimerkiksi decode_json_fields, drop_fields ja include_fields. Decode_json_fields avaa JSON-rakenteen ja muodostaa sen uudelleen halutun sisällön mukaiseksi. (Filebeat Docs 2017c.)

Drop_fields toiminnolla poistetaan kenttä ehtoon perustuen tai suoraan ilman ehtoja. Vastaavasti include_fields taas pitää kentän mukana ehtoon perustuen tai ilman ehtoa aina. (Filebeat Docs 2017c.)

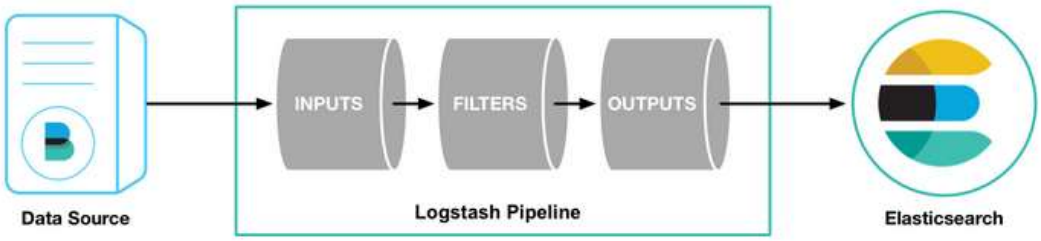
5.3.3 Useammalle riville jakautuva lokitieto

Mikäli yhden tapahtuman lokitieto on niin pitkä, että se ei sovi yhdelle riville tai yhdessä tapahtumassa käytetään rivinvaihtoa, vaatii Filebeat lisämäärityksiä. Multiline määrityksellä kerrotaan, mitkä tiedot kuuluvat yhteen tapahtumaan. Näin Filebeat kykenee käsittelemään jokaisen tapahtuman yhtenä kokonaisuutena. (Filebeat Docs 2017d.)

Lokitietoa lähetettäessä Logstashille on multiline-määritys ehdottoman tärkeä, muutoin data saattaa sekoittua ja korruptoitua. Mikäli kaikki tapahtuman lokitiedot ovat yhdellä rivillä, niin määritystä ei tarvita. (Filebeat Docs 2017d.)

5.4 Logstash

Logstash on putkimainen (pipeline) lokien käsittelytyökalu, joka pystyy ottamaan vastaan lokeja yhtä aikaa useasta eri lähteestä. Sen tehtävä on muokata lokit samaan muotoon. Sen toiminta jakaantuu kolmeen eri toiminnallisuuteen: sisääntulo (input), suodatus (filter) ja lähtö (output) (KUVIO 9). (Logstash 2017.)



KUVIO 9. Logstash toimintaperiaate (Logstash Docs 2017a)

Sisääntulossa määritetään, kuinka lokit otetaan vastaan Logstash-sovellukseen. Vaihtoehtoja voivat esimerkiksi olla hakemistorakenteessa sijaitseva tiedosto tai jossakin verkossa oleva laite. Suodatus-osiossa suodatetaan ylimääräinen tieto pois, sekä jäsennetään tieto haluttuun muotoon. Lähdössä määritetään lokien lähetys toisen ohjelman käsiteltäväksi. Sisääntulo ja lähtö ovat pakollisia osia. Suodatus-osio ei ole pakollinen, mutta sen käyttö auttaa ylimääräisen tiedon suodattamisessa ja kiinnostavan tiedon jäsentämisessä. Sisääntulo ja lähtö mahdollistavat koodekkien käytön, joiden avulla tieto voidaan koodata tai dekoodata haluttuun muotoon. (Logstash 2017; Logstash Docs 2017e.)

5.4.1 Logstash asetukset

Logstash-sovelluksella on kahden tyyppisiä asetuksia: pipelinen asetukset ja sovelluksen asetukset. Pipelinen asetuksilla hallitaan tiedonkulkua ja sovelluksen asetuksilla itse Logstash-sovelluksen käynnistystä ja toteutusta. (Logstash Docs 2017d.)

Logstash-pipelinen asetuksia hallitaan konfiguraatio tiedostojen avulla. Konfiguraatio tiedostoja voi olla yksi tai useampia. Kaikkien kolmen osioin asetukset voidaan sijoittaa yhteen tiedostoon tai jokainen osio voidaan sijoittaa erilliseen tiedostoon. Useata konfigurointitiedostoa käytettäessä, tiedoston nimen alkuun laitetaan järjestysnumero, jolla määritetään konfiguraatiotiedostojen suoritusjärjestys. Konfiguraatiotiedostojen sijainti

on /etc/logstash/conf.d. Konfiguraatiotiedostoissa käytetään päätettä ”.conf”. (Logstash Docs 2017b; Logstash Docs 2017d.)

Logstash-sovelluksen asetukset sijaitsevat /usr/share/logstash-kansioissa olevassa logstash.yml-tiedostossa. Asetustiedostoon määritetään esimerkiksi asetustiedostojen sijainti, käytettävät laajennukset, sovelluksen lokin kirjoittamisen tason ja pipeline-asetusten lataustapa. Useimmat asetuksista on käytettävissä myös komentoriviltä. (Logstash Docs 2017b; Logstash Docs 2017c.)

5.4.2 Sisääntulo

Sisääntulo-osioissa määritetään mistä ja missä muodossa tieto tuodaan Logstash-sovellukselle. Mahdollisia tietolähteitä voivat olla tiedosto(t), syslog, redis ja beats.

Tiedostojärjestelmässä olevaa tiedostoa luetaan file-toiminnon avulla. Tällöin määritetään tiedoston sijainti ja nimi. *-merkin käyttö on sallittua tiedostonnimeä ja sijaintia määritettäessä.

Syslog-toiminto mahdollistaa järjestelmä lokien lukemisen RFC3164 muodossa. Syslog-toiminnon käyttämä oletusportti on 514.

Redis-toiminto mahdollistaa tapahtumien lukemisen Redis-instanssista. Redis-toiminnossa on tuki kanavien ja listojen lukemiselle.

Beats on erillinen ohjelma, jonka tietovirtaa Logstash kuuntelee. Yksi esimerkki beat-sovelluksesta on Filebeat, joka on esitelty luvussa 5.3 Filebeat. (Logstash Docs 2017e.)

5.4.3 Suodatin

Suodatin on Logstash-putkilinjan keskimmäinen osa. Suodatin-osio on varsinainen Logstashin toiminnallinen osio. Kuten aikaisemmin on mainittu, siinä nimensä mukaisesti suodatetaan tietoa, mutta siellä pystytään myös muokkaamaan tieto tarvittavaan muotoon. Suodatin-

osiossa on mahdollista käyttää useita eri suodattimia ehtolauseita hyödyntäen. Seuraavassa on esitelty yleisimmin käytettyjä suodattimia. (Logstash Docs 2017e.)

Grok on jäsentämättömän tekstin suodatintoiminto. Se on tällä hetkellä paras suodatin rakenteettomien lokitiedostojen jäsentämiseen käsiteltävään muotoon. Grok-suodattimen toiminta perustuu ennalta määritettyjen merkkijonojen etsimiseen. Logstashin sisään on rakennettu 120 erilaista mallia Grog-suodattimen käyttöön. (Logstash Docs 2017e.)

Mutate on suodatin, jolla pystytään tekemään yleisiä muutoksia tekstitiedoston kentille. Esimerkiksi kenttiä voidaan uudelleen nimetä, poistaa, lisätä, korvata ja muokata Mutate-suodattimen eri toiminnolla. Mutate-suodatin mahdollistaa myös kenttien sisällön muokkaamisen. (Logstash Docs 2017e.)

Drop-suodattimella voidaan nimensä mukaisesti poistaa haluttuja kohteita lokitiedostosta. Clone-suodatin taas mahdollistaa kenttien kopioimisen. (Logstash Docs 2017e.)

Geoip-suodatin mahdollistaa IP-osoitteen perusteella maantieteellisen sijainnin esittämisen kartalla. Geoip-suodatin vaatii Kibana-sovelluksen tiedon esittämiseen. (Logstash Docs 2017e.)

JSON-suodatin jäsentää JSON-tiedoston kentän Logstashin vaatimaan muotoon. Mikäli JSON-tiedon jäsentäminen ei onnistu, suodatin palauttaa ”_jsonparsefailure”-ilmoituksen. (Logstash Docs 2017e; Logstash Docs 2017f.)

5.4.4 Lähtö

Lähtö-osio on Logstash-pipelinen viimeinen vaihe. Siinä nimensä mukaan määritellään Logstash-sovelluksen lähtö, eli se, minne jäsennetty data lähetetään. Yleisimmin käytettyjä Logstashin lähtömäärityksiä ovat Elasticsearch ja tiedosto. Erilaisilla lisäosilla pystytään kuitenkin lähettämään jäsennetty data moniin eri kohteisiin. Esimerkkisi email-

lisäosalla data voidaan lähettää suoraan sähköpostiin. (Logstash Docs 2017e.)

Elasticsearch-lähtö on määritys Elasticsearch-palvelimesta, jonne jäsennetty data lähetetään. Määrityksissä on vähintään kerrottava, missä IP-osoitteessa Elasticsearch sijaitsee ja mitä porttia se kuuntelee. Tiedosto määrityksellä data tallennetaan suoraan tiedostojärjestelmässä olevaan tiedostoon. (Logstash Docs 2017e.)

Käytettäessä Elasticsearch sovellusta Logstashin lähtö-osiossa määritetään myös Elasticsearch-sovelluksessa käytetty malli (template). Mallin sisältö esitellään luvussa 5.5.4 Elasticsearch mallien käyttö.

5.4.5 Lisäosat ja koodekit

Logstash lisäosat (plugin) ovat itsenäisiä paketteja, joilla saadaan tehtyä lisätoimintoja datalle. Lisäosat voivat olla sisääntulo, suodatus, koodekki ja lähtö osioissa. Järjestelmään asennettuja lisäosia voidaan tarkastella komennolla ”/usr/share/logstash/bin/logstash-plugin list”. Lisäosia voi asentaa, poistaa ja päivittää käyttäen Logstashin komentoriviä. (Logstash Docs 2017h.)

Koodekit ovat tietovirran suodattimia, jotka toimivat tarvittaessa Logstash-sovelluksessa osana tuloa ja lähtöä. JSON-suodatin mahdollistavat JSON-tiedostojen lukemisen ja lähettämisen. Multiline-määrityksellä mahdollistetaan usealle riville jakaantuvan yhden tapahtuman käsittely. (Logstash Docs 2017e; Logstash Docs 2017g.)

5.5 Elasticsearch

Elasticsearch on haku- ja analytiikkamoottori, joka muuntaa tiedon hauille optimoituun muotoon ja näin mahdollistaa nopean tiedon hakemisen. Se mahdollistaa suurien tietomäärien tallentamisen, hakemisen ja analysoinnin nopeasti ja lähes reaaliajassa. Lokikäytössä dataa on paljon,

joten on tärkeää, että hakeminen on nopeaa kriittisissäkin tilanteissa. (Elasticsearch Docs 2017a.)

Elasticsearch on tarpeen mukaan hyvin joustava ja skaalautuva. Sitä voidaan ajaa pienissä ympäristöissä ja testikäytössä jopa kannettavalla tietokoneella. Isoissa ympäristöissä se voidaan skaalata jopa tuhansiin palvelimiin. Elasticsearch voidaan integroida muihin sovelluksiin ja siksi sen käyttö on yleistä eri palveluissa. Tällaisia ovat esimerkiksi Wikipedia ja GitHub. (Gormley & Tong 2015a.)

5.5.1 Klusteri (Cluster) ja solmu (Node)

Klusteri on yhden tai useamman solmun muodostama ryhmä. Yhteen klusteriin kuuluvilla solmuilla on yksilöllinen klusterinimi, joka Elasticsearch-sovelluksessa on oletuksena ”elasticsearch”. Klusterinimi on tärkeä, koska sen perusteella solmu kuuluu tiettyyn klusteriin ja kaikki Elastic-sovellukset käyttävät samaa klusterinimeä. (Elasticsearch Docs 2017b.)

Jo yhden solmun verkko muodostaa klusterin. Erinimisiä klustereita voi olla useita. Klusteri mahdollistaa Elasticsearchin skaalautumisen tehovaatimuksiin sopivaksi, koska klusteriin voidaan joustavasti lisätä tai poistaa solmuja. Klusterissa on yksi isäntäkone, joka hallinnoi tietojen jakamista. (Elasticsearch Docs 2017b; Gormley & Tong 2015b.)

Solmu on yksittäinen palvelin, joka on osa klusteria. Solmun tehtävänä on tiedon tallentaminen, indeksointi ja hakeminen. Solmu tunnistetaan yksilöllisellä satunnaisella nimellä Universally Unique IDentifier (UUID). Solmun nimen voi tarvittaessa määrittää myös manuaalisesti. (Elasticsearch Docs 2017b.)

5.5.2 Indeksi (Index)

Indeksi on samankaltaisen tiedon kokoelma. Samankaltaisella tiedolla tarkoitetaan sitä, että tiedosta on eriteltävissä tarvittavilta osin

samantyyppinen tieto (samantyyppiset tietokentät). Koko tiedon ei tarvitse olla täysin saman tyyppistä. Riittää, kun käyttöön tulevat osat ovat vastaavia. (Elasticsearch Docs 2017b.)

Indeksi tunnistetaan yksilöllisellä pienin kirjaimin muodostetulla nimellä. Indeksinimi vaaditaan, kun Elasticsearch indeksoi tiedon tai Kibana esittää tiedon sisältöä. Logstash-sovelluksen lähtö-osioon määritetään indeksinimi. Yhteen klusteriin voi määrittää useita indeksejä. (Elasticsearch Docs 2017b.)

5.5.3 Indeksin jakaminen paloihin ja replikointi

Yhteen indeksiin voidaan tallentaa suuri määrä tietoa. Indeksin koko voi olla niin iso, että se ei mahdu yhden solmun levytilalle tai sen käyttö on liian hidasta. Tällöin joudutaan jakamaan indeksi paloihin usealle solmulle. Jokainen pala on tavallaan itsenäinen ”indeksi”, jota indeksin isäntäkone hallinnoi. Tallennuskapasiteetin kasvun lisäksi indeksin jakaminen lisää suorituskykyä, koska tietoa voidaan tallentaa tai hakea rinnakkain saman aikaisesti usealta solmulta. (Elasticsearch Docs 2017b.)

Toinen vaihtoehto suorituskyvyn lisäämiseen on saman tiedon replikointi usealle palvelimelle, jolloin päästään tiedon hakuvaiheessa hakemaan tietoa samanaikaisesti rinnakkain. Replikointi parantaa myös tiedon saatavuutta vikatilanteissa. (Elasticsearch Docs 2017b.)

5.5.4 Elasticsearch mallien käyttö

Mallien käyttö uuden indeksin luonnin yhteydessä on hyödyllistä, sillä mallit sisältävät indeksin asetuksiin ja rakenteeseen liittyvää tietoa. Esimerkiksi tietokenttien tyyppi voidaan määrittää mallissa. Malleja käytetään vain indeksiä luomisen yhteydessä. Mallin muokkaaminen ei vaikuta olemassa oleviin indekseihin. (Elasticsearch Docs 2017c.)

Useita malleja voidaan käyttää yhtä aikaa siten, että malleille määritetään järjestysnumero ja indeksinimeen perustuva ehto. Mallit suoritetaan

järjestysnumeron mukaisessa järjestyksessä. Indeksinimiehto voi olla esimerkiksi ”*”, jolloin mallia käytetään kaikille indekseille tai ”lo*”, jolloin mallia käytetään indekseihin joiden nimi alkaa lo-kirjaimilla. (Elasticsearch Docs 2017c.)

5.6 Kibana

Kibana on selainpohjainen analytiikka- ja visualisointialusta, joka on suunniteltu toimimaan yhdessä Elasticsearchin kanssa. Kibana mahdollistaa Elasticsearchin indeksoimien tietojen vuorovaikutteisen etsimisen, tarkastelun ja tallentamisen. Kibana mahdollistaa tietojen tarkastelun ja analysoinnin erilaisten taulukoiden, kaavioiden ja karttojen avulla tapahtuman aikaleimaan perustuen. (Kibana Docs 2017a.)

Elasticsearchin muodostama indeksi on Kibanassa indeksikuvio (Index Pattern). Esimerkki Kibanan käyttöliittymästä on esitetty alla olevassa kuviossa 10. (Kibana Docs 2017a.)

KUVIO 10. Kibana käyttöliittymä (Kibana, 2017)

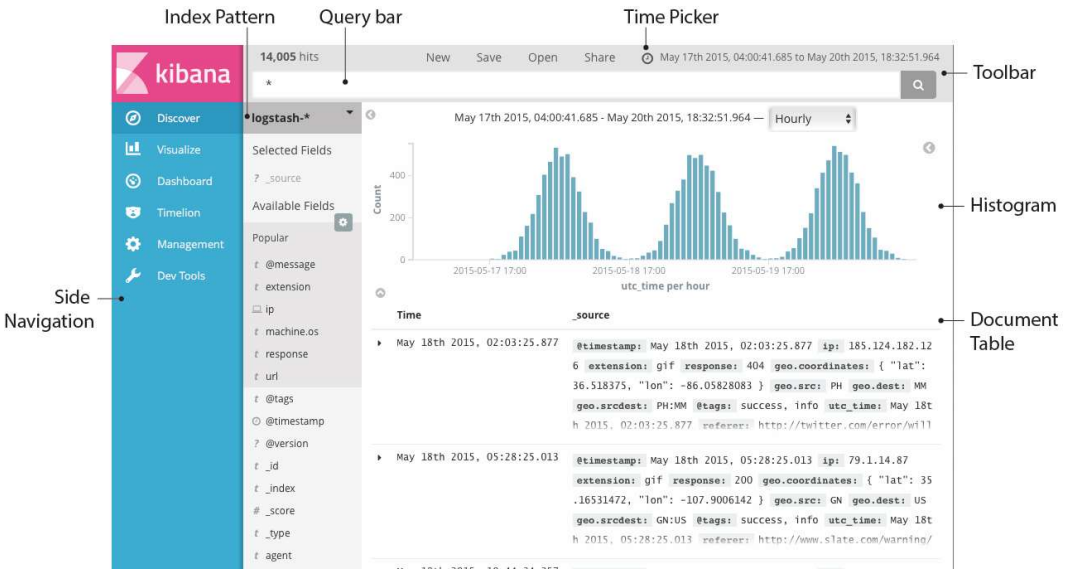
5.6.1 Kibanan käyttöliittymä

Perusasennetun Kibanan käyttöliittymä muodostuu kuudesta eri välilehdestä: Discover, Visualize, Dasboard, Timelion, Dev Tools ja Managerment. Discover-välilehdellä etsitään ja tarkastellaan käytettävissä olevia lokeja. Visualize-välilehdellä luodaan erilaisia kuvaajia. Dasboard-välilehdellä luodaan ja tarkastellaan useiden kuvaajien raportointinäkymiä.

Timelion-välilehdellä muodostetaan kuvaajia komentokielellä aikaleimaan perustuen. Dev Tools -välilehdellä hallitaan ja tarkastellaan Elasticsearchin indeksien sisältöä ja asetuksia komentorivipohjaisesti. Management välilehdellä luodaan ja hallitaan Indeksikuvioita ja määritetään Kibanan asetukset. (Kibana Docs 2017a; Kibana Docs 2017b.)

5.6.2 Discover

Discover-välilehti (kuvio 11) sisältää indeksikuviovalinnan (Index Pattern), aikasuodattimen (Time Picker), kyselyrivin (Query bar), histogrammin (Histogram) ja lokitiedon taulunäkymän (Document Table). Discover-välilehdellä nähdään valitun indeksin sisältö valitulta aikaväliltä. Histogrammi piirtää tapahtumamäärän kunkin aikaleiman kohdalle. Lokitietojen taulunäkymässä tarkastellaan lokitiedon sisältöä kyselyrivin komentojen perusteella. (Kibana Docs 2017c.)



KUVIO 11. Kibana Discover näkymä (Kibana Docs 2017c)

5.6.3 Visualize

Visualize-välilehdellä luodaan ja muokataan valitusta indeksistä kuvaajia. Mahdollisia kuvaajia ovat esimerkiksi erityyppiset pylväs- ja piirakkadiagrammit, taulukot ja karttanäkymät. Aiemmin luoduista kuvaajista muodostuu lista, jonka kuvaajia voidaan muokata ja tarkastella. (Kibana Docs 2017d.)

Kuvaajia luodessa on mahdollista hyödyntää Kibanaan rakennettuja laskutoimituksia. Näin ollen kentän sisällön ei tarvitse olla valmiiksi esittävä tietoa, vaan kentän tiedosta tehdään jokin laskutoimitus ja esitetään tulos kuvaajana. Mahdollisia laskutoimituksia kuvaajan muodostuksessa ovat esimerkiksi keskiarvo, summa, minimi, maksimi, prosentti ja derivaatta. (Kibana Docs 2017d.)

5.6.4 Dashboard

Raportointinäkymä on käyttäjälle näkyvä osa. Sen tehtävänä on näyttää käyttäjälle tärkeimmät tiedot yhdellä näytöllä, ohjata tietojen selaamisessa,

huolehtia mahdollisesta tietojen päivittämisestä näytölle ja suorittaa tarvittavia laskutoimituksia. (Yuk & Diamond 2014, 22.)

Kibanan Visualize-välilehdellä luoduista kuvaajista rakennetaan raportointinäkymän Dashboard-välilehdellä. Raportointinäkymään voidaan tuoda useita kuvaajia. Kuvaajien koko ja sijainti voidaan muokata tarpeen mukaan. (Kibana Docs 2017e.)

5.6.5 Timelion

Timelion-välilehdellä voidaan muodostaa kuvaajia nopeasti pelkkiä tekstipohjaisia komentoja hyödyntäen. Kuvaajien muodostuksessa voidaan hyödyntää laskutoimituksia ja aikaleimaa. Myös Timelion-välilehdellä muodostetut kuvaajat on mahdollista lisätä Dashboard välilehden näkymään. (Kibana Docs 2017f.)

Timelion-sovelluksella on mahdollista tehdä ehdollista logiikkaa, jos esittäminen sitä vaatii. Pidemmällä aikavälillä ehdollinen logiikka auttaa löytämään poikkeamia ja tiettyjä malleja lokiaineistosta. (Kibana Docs 2017f.)

5.6.6 Dev Tools

Dev Tools -välilehden konsoli tarjoaa tekstipohjaisen käyttöliittymän Elasticsearch-palveluun ja sen tekemiin indeksitietoihin REST-rajapinnan kautta. Konsolissa on käskyalue ja tulosten tarkastelualue. Käskyalueelle voidaan kopioida tai suoraan kirjottaa useita eri käskyjä allekkain ja käyttää niitä tarpeen mukaan. (Kibana Docs 2017g.)

Tulosten tarkastelualueelle muodostuu näkymä käskyn perusteella. Näkymä voilla olla esimerkkisi tieto komennon suorittamisesta tai näkymä indeksin sisällöstä. (Kibana Docs 2017g.)

5.6.7 Management

Management-välilehdellä hallitaan Kibana-sovelluksen asetuksia ja indeksikuvioiden asetuksia. Välilehdellä lisätään uudet indeksikuviot Kibanan muodostamien kuvaajien käyttöön. *-merkin käyttö indeksikuvion nimessä on sallittua, jolloin pystytään viittaamaan yhteen tai useampaan Elasticsearchin indeksiin yhdellä viittauksella. (Kibana Docs 2017h.)

Asetuksista on mahdollistaa muokata esimerkiksi kenttien tyyppiä ja aikaleiman muotoa. Lisäasetuksista päästään muokkaamaan Kibanan käyttäytymistä ja suorituskykyä. Tällaisia asetuksia ovat esimerkiksi Kibanan käyttämä aikavyöhyke ja kenttien näkymäasetukset. (Kibana Docs 2017h.)

5.7 X-Pack

X-Pack on Elastic Stackin laajennuspaketti, joka lisää työkaluja tietoturvaan, hälytyksiin, valvontaan, raportointiin, koneoppimiseen ja grafiikkaan. Käytön kannalta sen asentaminen ei ole pakollista, mutta siitä saa huomattavaa lisähyötyä. (X-Pack Docs 2017a.)

Tietoturvatyökalu auttaa klusterin suojauksessa. Sen avulla on mahdollista parantaa käyttäjien hallintaa, IP-suodatusta ja salata klusterin osien välisiä yhteyksi. Tietoturvatyökalu esimerkiksi mahdollistaa käyttäjätunnistuksen Active Directorystä ja roolipohjaisen kirjautumisen. Lisäksi siinä on työkalu tietojen eheyden varmistamiseen. (X-Pack Docs 2017e.)

Valvontatyökalu mahdollistaa Elastic Stack -sovellusten valvonnan. Sillä nähdään sovellusten tila ja vaatima teho sillä hetkellä sekä historiatietoa. Valvontatyökalu mahdollistaa yhden tai useamman klusterin valvomisen keskitetysti. (X-Pack Docs 2017f.) Hälytystyökalu mahdollistaa klusterin sovellusten tilan tarkkailun ja siitä hälyttämisen sekä indeksoitavan tiedon sisällön perusteella tehtävät hälytykset (X-Pack Docs 2017g).

Grafiikkatyökalulla on mahdollista piirtää kuvaajia tietojen välisistä suhteista. Kuvaajien avulla indeksoidusta lokitiedosta voidaan löytää uutta

tietoa, joka mahdollista esimerkiksi WEB-kaupan asiakkaille tuotteiden suosittelemisen. (X-Pack Docs 2017c.)

Raportointityökalulla voidaan luoda esimerkiksi Kibanan visualisoinneista PDF-tyyppisiä raportteja (X-Pack Docs 2017b). Koneoppimisen avulla pystytään luomaan normaalin toiminnan malleja ja tunnistamaan poikkeavuuksia (X-Pack Docs 2017h).

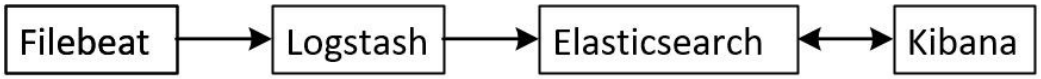
X-Pack on Elasticin maksullinen tuote. Asennuksen yhteydessä siitä saa 30 päivän testiversion. Rajoitetuilla ominaisuuksilla sitä on mahdollista käyttää myös testiajan jälkeen. Lisenssistä on olemassa neljää eri vaihtoehtoa: Basic, Gold, Platinum ja Enterprise. Lisenssitason kasvaessa lisääntyy myös X-Pack ominaisuuksien määrä sekä tukipalvelun nopeus paranee. (X-Pack Docs 2017d.)

6 KÄYTÄNNÖN TOTEUTUS

6.1 Toteutuksen suunnitelma

Elasticin dokumentaation ja sovellusten testikäytön perusteella työssä päädyttiin käyttämään Elasticin neljää sovellusta (KUVIO 12). Työn käytännön toteutus suoritetaan käyttäen Elasticin sovelluksia Filebeat, Logstash, Elasticsearch ja Kibana.

Filebeat sovelluksen tehtävänä on lähettää lokitieto Logstash-sovellukselle. Logstash suodattaa lokitiedosta ylimääräisen tiedon pois ja muokkaa tarvittavan tiedon samaan haluttuun muotoon. Muokatut tiedot Logstash tallentaa Elasticsearchiin, josta Kibana lukee ne käyttöliittymälle. Kibana tekee myös tallennuksia Elasticsearchiin. Esimerkiksi kaikki Kibanan kuvaajat ja raportointinäkymät tallennetaan yhteen indeksiin Elasticsearchissä.



KUVIO 12. Käytetyt sovellukset

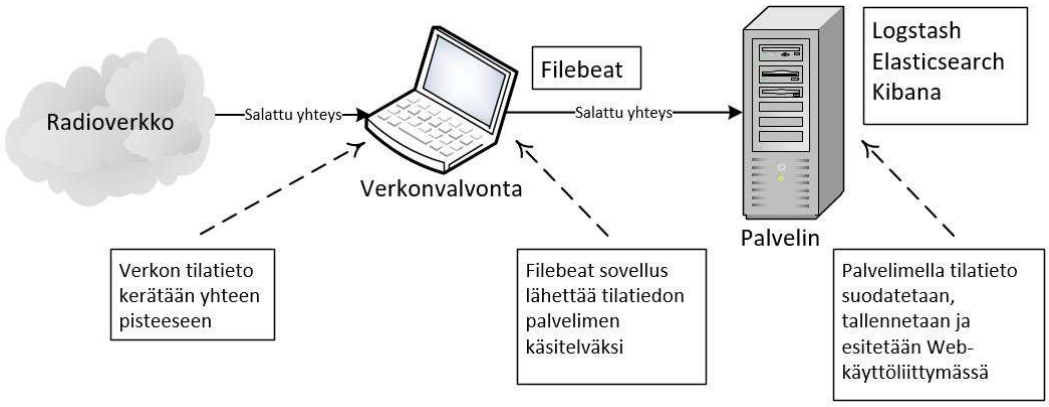
Vaikka sovelluksia on paljon, niillä jokaisella on toteutuksen kannalta oma roolinsa. Mikäli käytettäisiin Filebeatin suodatusominaisuuksia, Logstash olisi mahdollista jättää pois, mutta tällöin suodatuksen aiheuttama kuormitus muodostuisi lokeja tuottavalle tietokoneelle eikä varsinaiselle palvelintietokoneelle. Lisäksi Logstashin suodatinominaisuudet ovat laajemmat ja kehittyneemmät kuin Filebeatin. Muita sovelluksia ei ole mahdollista jättää pois. Tästä syystä kaikki neljä sovellusta tarvitaan työn toteutukseen.

Elastic-sovelluksiin päädyttiin, koska valmistajan internetsivuilla on laajadokumentaatio kaikista sovelluksista. Sovelluksiin saa tukea

keskustelupalstoilta, joissa on valmiita esimerkkejä ja ratkaistuja ongelmatilanteisiin. Lisäksi sivustolla discuss.elastic.co on mahdollista avata tukipyyntö ongelma- tai vikatilanteesta. Valmiita asetusesimerkkejä löytyy myös usealta sivustolta, joka mahdollistaa aloittelijalle alkuun pääsemisen ja hankalampienkin tilanteiden toteutuksen.

6.2 Järjestelyjen yleiskuvaus

Järjestelyjen yleiskuvaus (KUVIO 13) on esitetty seuraavassa kuviossa. Filebeat-sovellus asennettiin verkonvalvontatietokoneeseen, johon radioverkon tilatiedoista koostuva lokitieto muodostuu. Verkonvalvontasovellus tallentaa lokitiedon tiettyyn sijaintiin verkonvalvontatietokoneella. Filebeat sovellus lähettää lokitiedon palvelimella sijaitsevalle Logstashille sitä mukaa, kun lokitietoa syntyy.



KUVIO 13. Järjestelyjen yleiskuvaus

Palvelimella sijaitseva Kibana tuottaa Web-käyttöliittymän (raportointinäkymä) tietojen selaamiseen. Tällöin tietoja on mahdollista tarkastella kaikkialta verkosta salatun yhteyden yli.

Tässä versiossa kaikki lokitiedon käsittelyyn liittyvät sovellukset on asennettu samaan palvelintietokoneeseen. Kaikki sovellukset mahdollistavat myös sen, että ne asennetaan eri palvelimille, jolloin

järjestelmän aiheuttama kuormitus saadaan jaettua useammalle palvelimelle. Tällä hetkellä tähän ei kuitenkaan ole nähty mitään tarvetta.

6.3 Elastic-sovellusten käyttöönotto

Kaikki työssä käytetyt Elasticin sovellukset asennettiin käyttäen Linux-käyttöjärjestelmän apt get -paketin hallintaa. Ohjeet kunkin sovelluksen asentamisesta löytyvät Elasticin sivulta kyseisen sovelluksen kohdalta.

Työn toteutuksen aikana sovelluksiin julkaistiin useita päivityksiä. Päivitykset asennettiin käyttäen apt-get paketin hallintaa. Sovellusten asentamisessa tai päivittämisessä ei havaittu ongelmia. Kaikki neljä sovellusta asetettiin käynnistymään automaattisesti käyttöjärjestelmän käynnistymisen yhteydessä.

6.4 Filebeat käyttö

Filebeat lähettää kaiken lokitiedon palvelimen käsiteltäväksi sitä mukaa, kun sitä muodostuu. Se ei suodata tai muutoinkaan käsittele tietoa millään tavalla vaan lähettää ainoastaan tiedon eteenpäin. Filebeat-sovelluksessa on mahdollista tehdä datan suodatusta. Tässä tilanteessa se ei ole kuitenkaan järkevää, koska Filebeat sovelluksen aiheuttama kuormitus verkonvalvontakoneella on pidettävä mahdollisimman alhaisena. Suodatuksen tekeminen valvontatietokeella ei ole muutoinkaan tarpeellista, koska tietokoneiden välillä on nopea lähiverkko ja vastapäässä on tehokas palvelintietokone. Näin ollen suurin työ tehdään palvelimella.

Filebeat asetukset sijaitsevat tiedostossa filebeat.yml (KUVIO 14). Tiedosto sijaitsee kansiossa /etc/filebeat/.

```
filebeat.prospectors:
- input_type: log
  paths:
    - /var/log/*.json
output.logstash:
  hosts: ["XXX.XXX.XXX.XXX:5043"]
```

KUVIO 14. Filebeat asetukset

Filebeat lukee kaikki JSON-tyyppiset tiedostot kansiosta /var/log. Sovelluksen rekisteri huolehtii siitä, että kaikki data luetaan vain kerran, vaikka Filebeat käynnistettäisiin uudelleen. Tiedostoihin kirjoitetut uudet lokitiedot luetaan sitä mukaa, kun niitä syntyy.

Kaikki Filebeat-sovelluksen lukemat lokitiedot lähetetään Logstash-palvelimelle. Logstash-palvelimen IP-osoite on kuvattu merkein XXX.XXX.XXX.XXX. Logstash-palvelun sisääntuloportiksi on määritetty 5043.

Filebeat-sovelluksen virhelokit tulostuvat tiedostoon /var/log/logstash/filebeat. Vanhat lokitiedot tallennetaan samaan kansioon juoksevalla numerolla lisättynä.

6.5 Logstash käyttö

Logstash-sovelluksen käyttöön on kaksi vaihtoehtoa. Käytetään automaattisesti käynnistynyttä Logstash-palvelua tai suoritetaan Logstash-palvelu komentoriviltä. Käytettäessä automaattisesti käynnistynyttä palvelua, on sovelluksen asetuksista (logstash.yml-tiedosto) otettava käyttöön automaattinen asetustiedoston uudelleen suorittaminen. Asetuksen nimi on ”config.reload.automatic”. Tällöin suodatusasetuksia muutettaessa ne ladataan käyttöön tietyin aikavälein. Tässä vaihtoehdossa pipelinen asetuskansiossa (/etc/logstash/conf.d) ei saa olla vanhoja tai ylimääräisiä asetustiedostoja, koska kaikki kansiossa olevat tiedostot suoritetaan.

Suoritettaessa Logstash-palvelu komentoriviltä, on kerrottava käytettävän pipeline-asetustiedon nimi. Tällöin terminaaliin tulostuu kaikki tapahtumat ja virheilmoitukset. Siksi tämän tavan käyttö koettiin paremmaksi sopivien asetusten etsintävaiheessa. Tällöin on kuitenkin huomioitava, että automaattisesti käynnistynyt Logstash-palvelu on sammutettava ennen komentorivikäyttöä. Varsinaisessa käytössä käytetään kuitenkin vain automaattisesti käynnistynyttä palvelua.

Logstashin virhelokit tulostuvat tiedostoon /var/log/logstash/logstash-plain.log. Vanhat lokitiedot tallennetaan samaan kansioon päivämäärätarkennuksella lisättynä. Logstashin lokitieto on tarpeellinen työkalu varsinkin sopivien asetusten etsintävaiheessa.

Työn toteutuksen kannalta Logstash-sovellus muodostui keskeisimmäksi ja selkeästi muita sovelluksia työläämmäksi. Kuten Logstash teoriaosuudessa on kerrottu, Logstash sovelluksen toiminta jakautuu kolmeen eri osioon: sisääntulo, suodatus ja lähtö. Näistä vaiheista sopivien suodatusasetuksien ja suodatettavien asioiden löytäminen vei selkeästi eniten aikaa. Yhtenä tekijänä vaikutti kokemattomuus Logstash-sovelluksesta. Toisena tekijänä oli todella laaja lokitieto, jonka rakenteen ymmärtäminen oli huomattavan hankalaa.

6.5.1 Lokitiedoston sisältö

Lokitieto muodostuu aikaleiman alle tallentuvasta koko radioverkon tilatiedosta. Tilatietoon tallennetaan kaikkien radioverkossa olevien laitteiden väliset yhteydet ja yhteyksien laatua kuvaavat arvot, virheilmoitukset ja paljon muuta statistiikkaa. Kaikki radioyhteyksien laatua kuvaavat arvot tallentuvat kahteen kertaan, koska tieto tallennetaan yhteysvälin kummankin pään laitteiden kannalta.

Yhden aikaleiman alle tallentuvan tiedon määrä vaihtelee siis radioverkon laitteiden määrän mukaan. Yhden aikaleiman alla voi olla esimerkiksi noin 115 000 merkkiä. Tiedot tallentuvat sisäkkäiseen rakenteeseen siten, että syvyyssuunnassa voi sisäkkäisiä kenttiä olla jopa seitsemässä tasossa.

Tarkkaa tietoa kenttien määrästä ei ole, mutta se on yli 5000 kenttää. Kokonaiskenttämäärätietoa ei saatu selville, koska yhtä Elasticsearch-palvelinta käytettäessä näin suuren kenttämäärän indeksointi ei onnistu. Oletuksena Elasticsearch pystyy indeksoimaan korkeintaan 1000 kenttää. Kenttämäärä voidaan asetuksista säätää isommaksi, mutta tällöin ongelmaksi muodostuu koneteho, joka loppuu kesken yritettäessä indeksoida kaikkia kenttiä. Kokonaiskenttämäärä ei ole kuitenkaan olennaista tietoa, kun varsinaisessa työn toteutuksessa tarvittiin noin 400 kenttää.

6.5.2 Logstash pipeline-asetukset

Logstash pipeline asetuksista tehtiin viisi asetustiedostoa. Yhdessä tiedostossa on sisääntuloasetukset, kolmessa erillisessä tiedostossa on varsinaiset suodatusasetukset ja viimeisessä tiedostossa on lähtöasetukset. Asetukset jaettiin useaan tiedostoon käsiteltävyyden ja ymmärrettävyyden vuoksi. Tiedostot on nimetty siten, että niiden nimi alkaa suoritusjärjestykseen viittaavalla numerolla. Sisääntulo on 01, suodatukset ovat 10-12 ja lähtö on 20. Esimerkiksi ensimmäinen suodatusasetustiedosto on 10_filter_mutate.conf. Logstash suodatusasetukset sijaitsevat kansiossa /etc/logstash/conf.d.

6.5.3 Logstash sisääntulo

Filebeat-sovellukselta tulevat lokitiedot otetaan vastaan Logstashin (KUVIO 15). Asetukset sijaitsevat tiedostossa 01_input_filebeat.conf. Asetuksissa määritetään sisääntuloportti, joka on oltava sama, kuin Filebeat asetuksissa määritetty. Lokitietoon lisätään network-merkintä, jos tietoja joudutaan ottamaan vastaan useammasta lähteestä ja suodattamaan sen perusteella. Tiedon tyypiksi on määritelty JSON, joka on alkuperäisen lokitiedon tyyppi.

```
input {
   beats {
        port => "5043"
        tags => "network"
        codec => "json"
    }
}
```

KUVIO 15. Logstash sisääntuloasetukset

6.5.4 Logstash suodatukset

Lokitiedoston sisällön ollessa todella laaja, korostuu siitä ylimääräisen tiedon poistaminen. Suodatusasetusten löytäminen ja rakentaminen oli työssä keskeisessä roolissa. Suodatinasetuksilla poistetaan kaikki ylimääräinen tieto ja vain tarvittavat tieto jätetään. JSON-muotoisessa lokitiedostosta voidaan poistaa ylimääräiset tiedot kenttänimien perusteella. Suodatuksia rakennettaessa on ensin selvitettävä poistettavat kenttänimet ja niiden muodostama sisäkkäinen rakenne.

JSON-lokitiedoston rakenteen selvittämiseen käytettiin apuna Notepad++-editoria, koska sillä on mahdollista selvittään vastinparien välinen sisältö. Esimerkiksi aaltosulkujen {} väliin jäävä alue. Tämä helpotti alustavaa rakenteen selvittämistä ja ylimääräisen tiedon poistamista Logstash suodattimilla. Notepad++-editorilla päästiin hyvään alkuun, mutta sillä on

mahdotonta selvittää lokitiedoston koko rakennetta, kun sisäkkäisiä tasoja on useita ja niiden sisällä on useita kenttiä. Suodatinasetusten tekemiseksi tarvittavien kenttien ja niiden muodostaman rakenteen selvittäminen oli kuitenkin välttämätöntä.

Lokitiedoston rakenne selvisi kuitenkin vasta sen jälkeen, kun ensimmäinen Elasticsearch indeksointi onnistui ja kenttärakenne oli saatu ladattua Kibanan käyttöliittymälle. Ennen ensimmäisen indeksoinnin onnistumista, lokitiedostosta jouduttiin poistamaan jokaisen aikaleiman kohdalta useita tuhansia kenttiä. Tässä hyödynnettiin Notepad++-editorilla selvitettyä kenttärakennetta. Indeksointiongelma on kuvattu tarkemmin kappaleessa 6.5.1 Lokitiedoston sisältö.

Lokitiedoston kenttärakenteen selvittyä päästiin rakentamaan varsinaisia suodatusasetuksia. Suodatuksen periaate on, että tarvittava lokitieto kopioidaan uuteen kenttärakenteeseen ja koko alkuperäinen lokitieto poistetaan. Alkuperäisessä kenttärakenteessa oli sisäkkäisiä kenttiä seitsemässä tasossa. Uudessa rakenteessa sisäkkäisiä kenttiä on vain kahdessa tasossa. Uusi rakenne helpottaa huomattavasi tietojen käsittelyä Kibanassa.

Modulaatiokuvaajien piirtämiseksi joudutaan tallentamaan kaksi tietoa: alkuperäinen modulaatio ja modulaatiota vastaava kokonaisluku. Pylväsdiagrammia ei voi piirtää teksti- ja numeroarvojen yhdistelmästä. Modulaatio arvo voi olla esimerkiksi 64QAM5/6. Kaikki muu kuvaajiin tuleva tieto voidaan kopioida suoraan uuteen rakenteeseen.

6.5.5 Logstash suodatusasetusten rakentaminen

Logstash suodatusasetuksissa toistuu sama rakenne useita kertoja siten, että vain osa tiedoista muuttuu. Suodatusasetusrivejä toteutettiin yhteensä noin 800. Näiden tekemisessä hyödynnettiin Linux-skriptejä. Skriptien käyttö muodostuu kolmesta eri tiedostosta: template, laitetieto ja suoritettava komentoskripti.

Template-tiedosto sisältää suodatinasetusten rungon. Laitetiedoista löytyy radioverkon laitteiden nimi ja IP-osoite. Skriptiä suoritettaessa sijoitetaan suodatinasetusten muuttuviin tietoihin laitteiden nimet ja IP-osoitteet.

Tämä työkalu helpottaa suodatinasetusten tekemistä ja muokkaamista merkittävästi. Muutoin suodatinasetusten teko olisi täysin käsityötä ja rivien suuresta määrästä johtuen todella työlästä ja hidasta. Näin vältetään myös kirjoitus- ja logiikkavirheet, jotka pahimmillaan pääsisivät Kibanan käyttöliittymälle asti.

Esimerkiksi paha logiikkavirhe voisi olla sellainen, että johonkin kenttään sijoitettaisiin aivan toisen kentän tiedot. Tällainen järjestelmällinen virhe selviäisi vasta tarkassa tietojen vertailussa, jossa alkuperäistä verkonvalvontakuvaa ja nyt tehtyjä kuvaajia verrataan vierekkäin kaikkien tietojen osalta. Tämä on toki työ, joka pitää ennen varsinaista käyttöönottoa tehdä.

6.5.6 Ensimmäinen Logstash suodatin

Ensimmäisen Logstash suodattimen asetuksissa (10_filter_mutate.conf) määritetään aikaleima ja luodaan kaikista käyttöön tulevista tiedoista uusi kenttä (KUVIO 16). Kuviossa on esitetty periaate ensimmäisen suodattimen asetuksista. Suodattimella-date määritetään lokitiedoston aikaleima (timestamp) Kibanalle sopivaan muotoon. Aikaleiman saaminen Kibanaan on välttämätöntä, jotta kuvaajia pystytään selaamaan aikaan perustuen.

Kuviossa 16 on esitetty periaate lokitiedon rakenteessa. Lokitieto rakentuu sisäkkäisistä tietokentistä siten, että tietokentän ”kentta1” alla on syvyyssuunnassa tietokenttiä seitsemässä tasossa. Ensimmäisenä luodaan ”uusikenttä” nimeltään ”uusikentta1.rx_cinr”, johon sijoitetaan kentta1 alla olevan rx_cinr-kentän tiedot. Uuden kentän luomiseen ja siihen tiedon sijoittamiseen käytetään mutate-suodattimen add_field-toimintoa.

```
# basic filter and mutate add_field
filter{

date{
 match => ["timestamp", "YYYY-MM-dd HH:mm:ss"]
}

mutate{
        add_field => {

"uusikentta1.rx_cinr" => "%{[kentta1][alikentta1.1][alikentta1.1...n][rx_cinr]}"
"uusikentta1.tx_cinr" => "%{[kentta1][alikentta1.1][alikentta1.1...n][tx_cinr]}"
"uusikentta1.rx_rssi" => "%{[kentta1][alikentta1.1][alikentta1.1...n][rx_rssi]}"
"uusikentta1.tx_rssi" => "%{[kentta1][alikentta1.1][alikentta1.1...n][tx_rssi]}"
```

KUVIO 16. Logstash suodatusasetukset 10_filter_mutate.conf

6.5.7 Toinen Logstash suodatin

Toisen Logstash suodattimen asetuksissa (11_filter_trasnlate.conf) luodaan modulaatiota vastaava kokonaisluku uuteen kenttään ja lopuksi poistetaan kaikki vastaanotetut lokitiedot päätasolta alkaen (KUVIO 17). Kuviossa on esitetty periaate toisen suodattimen asetuksista. Translate-suodattimella luodaan modulaatiota vastaava kokonaisluku. Translate-suodattimelle määritellään kolme tietoa: lähdekenttä, kohdekenttä ja kirjasto, jonka tiedoilla muunnos toteutetaan. Lähdekenttä määritetään field-toiminnolla. Kohdekentäksi määritetään uusi kenttä destination-toiminnolla. Kirjasto määritetään dictionary-toiminnolla. Kirjaston tietoa hyödynnetään itse käännöksessä. Kirjastoon on määritettävä kaikki järjestelmän modulaatiot ja niitä vastaavat kokonaisluvut.

Esimerkkiasetusten lopussa on mutate-suodattimen remove_field-toiminto, jolla poistetaan kaikki vastaan otetut lokitiedot päätasoviittauksella. Kentta1 ja kentta2 kuvaavat kahta päätason kenttää, joiden mukana poistetaan myös kaikki alitason kentät.

```
#translate and remove
filter{

translate {
        field => "[kentta1][alikentta1.1][alikentta1.1...n][rx_modulation]"
        destination => "uusikentta1.RXmodulation_num"
        dictionary => [ "64QAM5/6", "6", "64QAM3/4", "5", ... ]
        }
translate {
        field => "[kentta1][alikentta1.1][alikentta1.1...n][tx_modulation]"
        destination => "uusikentta1.TXmodulation_num"
        dictionary => [ "64QAM5/6", "6", "64QAM3/4", "5", ... ]
        }
.
.
.
mutate{remove_field => ["kentta1", "kentta2", ... ]}
}
```

KUVIO 17. Logstash suodatusasetukset 11_filter_trasnlate.conf

6.5.8 Kolmas Logstash suodatin

Kolmannen Logstash suodattimen asetuksissa (12_ filter_convert.conf) asetetaan kaikille uusille kentille oikea tietotyyppi: integer tai string (KUVIO 18). Kuviossa on esitetty periaate kolmannen suodattimen asetuksista. Asettamiseen käytetään mutate-suodattimen convert-toimintoa.

```
#mutate convert

filter{
mutate{convert => {

"uusikentta1.rx_cinr" => "integer"
"uusikentta1.tx_cinr"  => "integer"
"uusikentta1.RXmodulation_num" => "integer"
"uusikentta1.TXmodulation_num" => "integer"
"uusikentta1.rx_modulation" => "string"
"uusikentta1.tx_modulation" => "string"
.
.
.
}
```

KUVIO 18. Logstash suodatusasetukset 12_ filter_convert.conf

6.5.9 Logstash lähtö

Logstashin lähtöasetukset (KUVIO 19) määritetään kolme erillistä lähtöä. Ylimpänä on virheen käsittely. Mikäli tiedon käsittelyssä tapahtuu jokin virhe, siitä tehdään merkintä jsonparsefailure-lokitiedostoon. Toisena lähtönä on stdout, joka tulostaa kaikki tapahtumat näytölle.

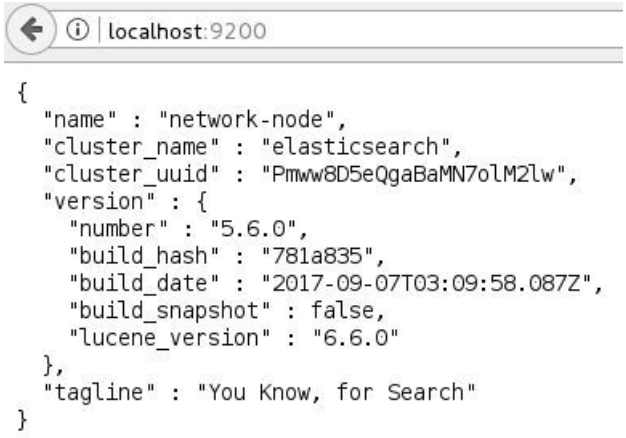
Kolmantena ja tärkeimpänä on määritetty Elasticsearch-lähtö. Asetuksissa kerrotaan, missä Elasticsearch-palvelin sijaitsee ja mitä porttia se kuuntelee. Lisäksi määritetään uusien indeksien nimi. Indeksinimen alkuosa muodostuu kiinteästä osasta ”logstash-”. Indeksinimen loppuosa taas muodostetaan lokitiedon aikaleiman päivämäärätiedosta. Näin jokaisen päivän lokitiedot tallentuvat omaan indeksiin, mutta niissä on myös saman sisältöinen osa. Tällainen rakenne helpottaa indeksien käsittelyä myöhemmin, koska niitä voidaan käsitellä yksittäisinä tai kaikkia kerralla. Tässä määritetty indeksinimi näkyy Elasticsearch- ja Kibana-sovelluksissa.

```
output {
          if  "_jsonparsefailure" in [tags] {
        file {
          path => "/var/log/logstash/jsonparsefailure.debug.log"
          codec => "rubydebug"
        }
      }
    stdout {
        codec => rubydebug
    }
    elasticsearch {
        hosts => "localhost:9200"
        index => "logstash-%{+YYYY.MM.dd}"
}
}
```

KUVIO 19. Logstash lähtöasetukset

6.6 Elasticsearch käyttö

Asennuksen jälkeen Elasticsearch ei vaadi asetusmuutoksia. Riittää, kun sovelluksen käynnistää odottamaan Logstashin lähettämiä lokitietoja ja Kibanan pyyntöjä indeksien sisällön esittämisestä. Elasticsearch-sovelluksen tiedot voi tarkastaa palvelintietokoneella osoitteesta http://localhost:9200/ (KUVIO 20).



KUVIO 20. Elasticsearch-asetukset

Elasticsearchin virhelokit tulostuvat tiedostoon /var/log/elasticsearch/ elasticsearch.log. Vanhat lokitiedot tallennetaan samaan kansioon päivämäärätarkennuksella lisättynä. Elasticsearchin virhelokin merkitys korostui työn alkuvaiheessa, kun varsinaista lokitietoa indeksoitiin ensimmäisiä kertoja.

Elasticsearchin tilaa, tapahtumia, klusterin ja solmujen tietoja voidaan tarkastella REST-rajapinnan kautta tai curl-komennoilla palvelimen terminaalista. Valmista REST-rajapintaa hyödyntävä käyttöliittymä löytyy Kibanasta kohdasta Dev tools. Työssä hallintaan käytettiin pääasiassa Kibanan REST-rajapintaa, mutta joissakin tilanteissa käytettiin myös curl-komentoja.

Seuraavana on lueteltu muutamat käytön kannalta tarpeellisimmat curl-komennot ja niitä vastaavat REST-rajapinnan komennot.

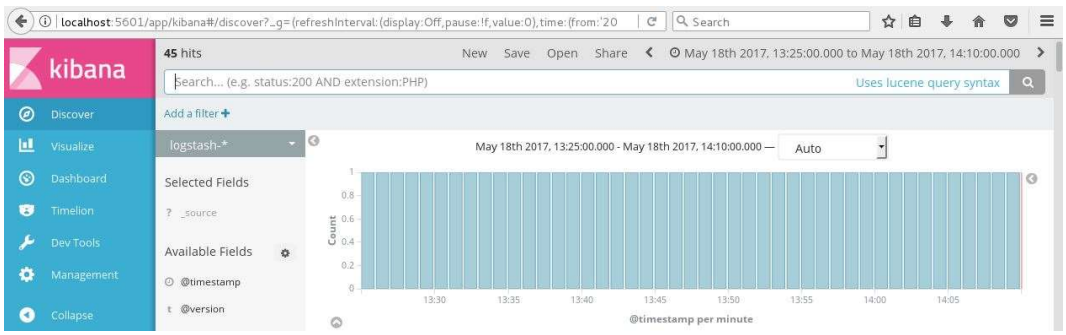
- Listaus klusterin tiedoista: curl 'localhost:9200/_cat/health?v'
- Clusterin indeksilistaus: curl 'localhost:9200/_cat/indices?v'
- Indeksin poisto: curl -XDELETE 'localhost:9200/”poistettavan indeksin nimi”?pretty'
- Clusterin solmujen listaus ja tietojen tarkastelu: curl 'localhost:9200/_cat/nodes?v'.

REST-rajapinnan komennot ovat

- Listaus klusterin tiedoista: GET /_cat/ health?v
- Clusterin indeksilistaus: GET /_cat/indices?v
- Indeksin poisto: DELETE /”poistettavan indeksin nimi”?pretty
- Clusterin solmujen listaus ja tietojen tarkastelu: GET /_cat/nodes?v

6.7 Kibanan käyttö

Logstash suodatuksen ja Elasticsearch indeksoinnin jälkeen lokitiedot ovat saatavana Kibanan käyttöön. Kibanan käyttöliittymään pääsee selaimella. Oletusasetuksilla käynnistetty Kibana-palvelin on samalta tietokoneelta käytettäessä osoitteessa http://localhost:5601 (KUVIO 21).



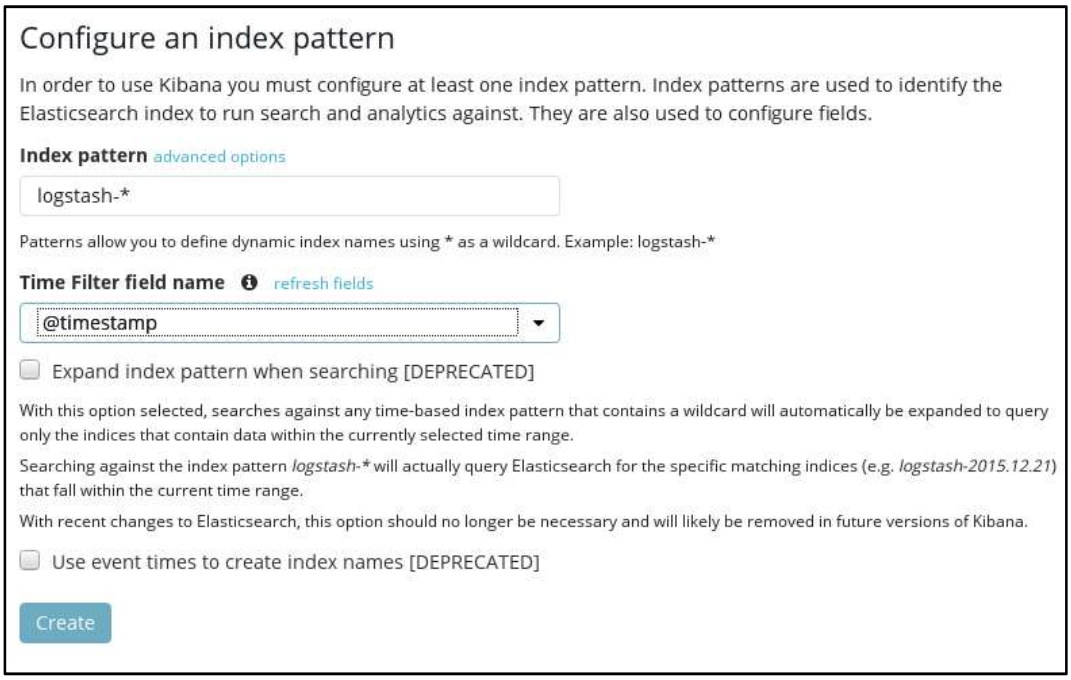
KUVIO 21. Kibanan käyttöliittymä

Raportointinäkymän tekeminen on yksinkertaistettuna kolmivaiheinen. Ensin Kibanalle luodaan indeksikuvio Elasticsearchin tekemästä lokitiedon indeksistä. Toisena luodaan indeksikuvion sisällöstä halutut kuvaajat ja kolmantena halutuista kuvaajista muodostetaan raportointinäkymät.

6.7.1 Kibana indeksikuvio

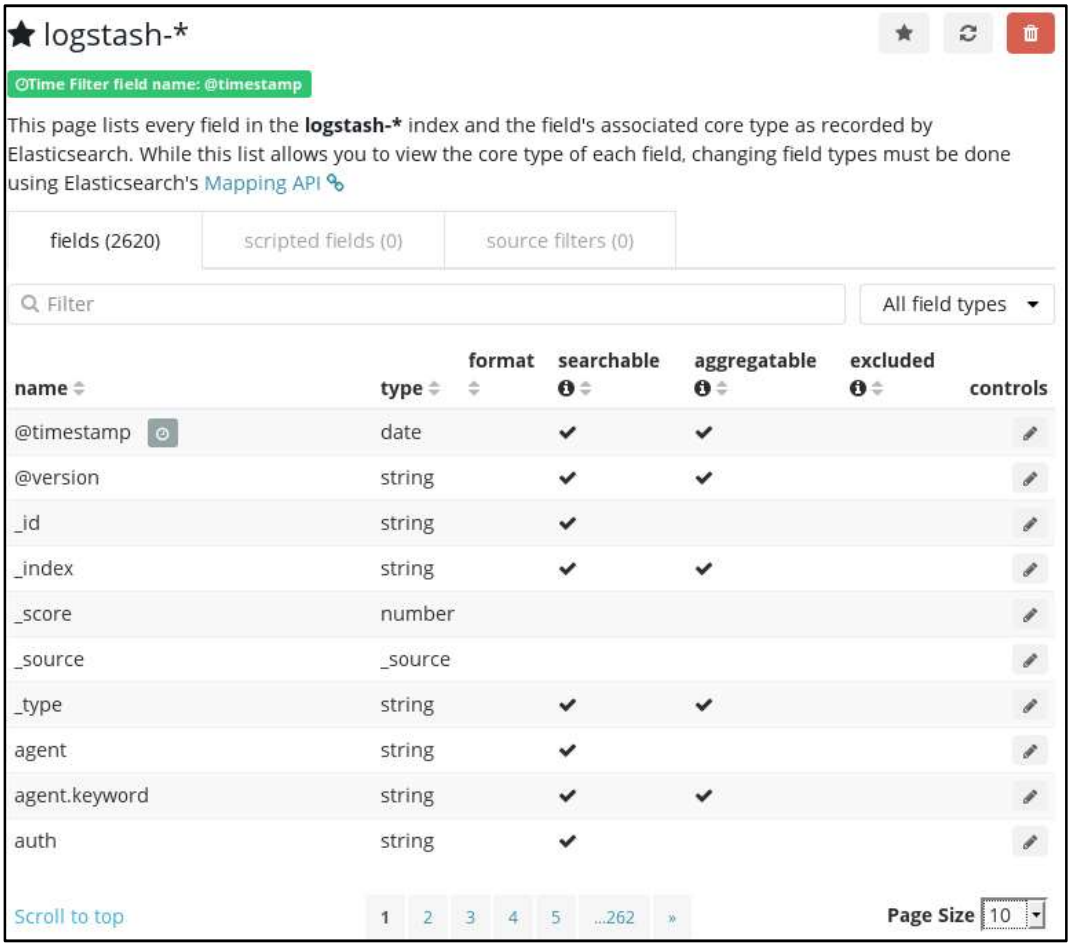
Kibana löytää Elasticsearchin indeksoimat tiedot indeksinimen perusteella, joka on määritelty Logstash-lähtöasetuksissa. Indeksinimestä muodostetaan Kibanassa indeksikuvio, johon tietyn indeksin sisältö haetaan. Indeksikuvionimi on käsiteltävän tietokokonaisuuden nimi.

Kibanan Management-välilehdellä, kohdassa Index Pattern on painike ”+Create Index”. Painikkeen valinnan jälkeen avautuu uusi sivu Index Pattern -asetusten määrittämiseksi (KUVIO 22).



KUVIO 22. Index Pattern -asetukset

Kibanan käyttämä indeksikuvion oletusnimi on logstash-*, joka mahdollistaa kaikkien Logstashin lähettämien indeksien näyttämisen Kibanalla. Logstashissa vastaava määritys on index => ”logstash-%{+YYYY.MM.dd}”. Päivämäärätiedon vaatimuksena on, että aikaleimakentän (timestamp) tieto määritetään Logstash-suodatinasetuksessa. Create-valinnan jälkeen päästään tarkastelemaan Indeksin sisältöä (KUVIO 23). Näkymässä nähdään kuinka paljon kenttiä indeksi sisältää, sekä jokaisen kentän nimi ja kentän tiedot.

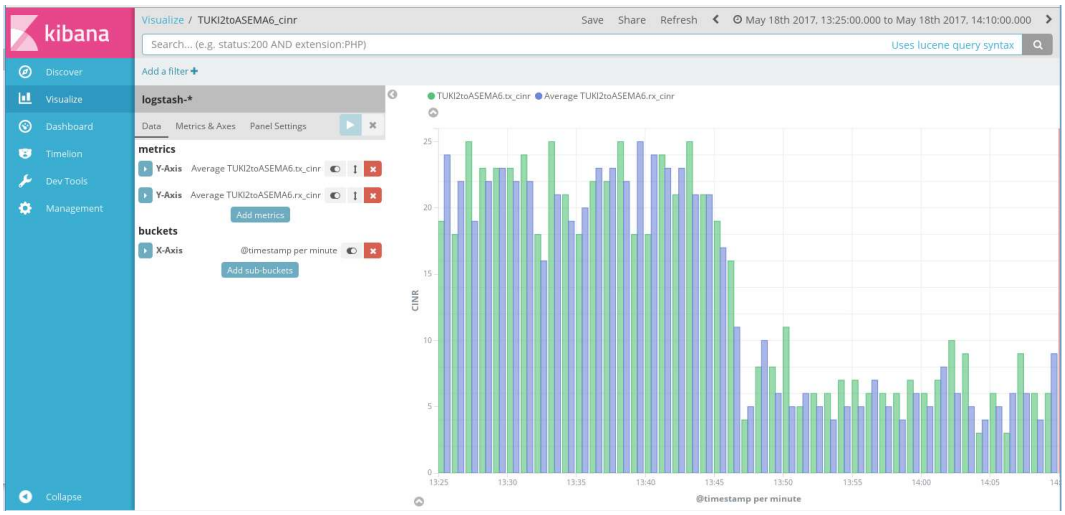


KUVIO 23. Indeksikuvion sisältö

6.7.2 Kibana Visualize

Kibanan Visualize-välilehdellä luodaan tarvittavat kuvaajat. Ensin valitaan kuvaajan tyyppi. Työssä käytettiin Vertical Bar -kuvaajia. Kuvaajan tyypin valinnan jälkeen avautuu ikkuna, jossa valitaan indeksikuvio, josta kuvaajia halutaan piirtää. Indeksikuvion nimeksi valitaan logstash-*. Seuraavana valitaan X- ja Y-akseleille tulevat kentät, kenttien asetukset ja otsikkotiedot (KUVIO 24). Työssä kaikkien kuvaajien X-akselille valittiin aikaleimakenttä ja Y-akselille RX- ja TX-radioarvot kuvaajan sisällön mukaan.

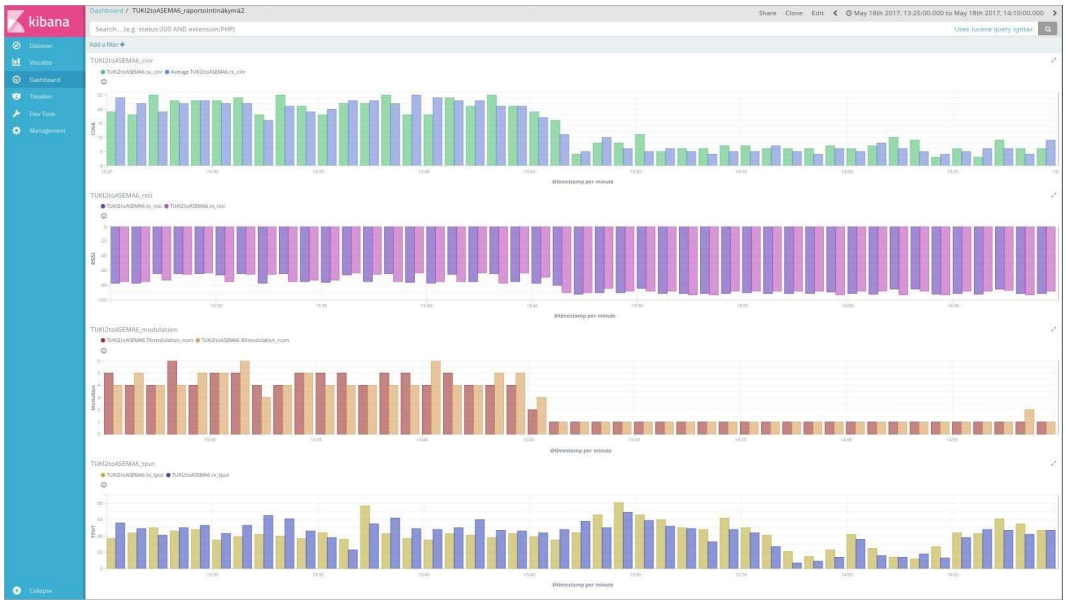
Tässä vaiheessa määritetään vielä pylväsdiagrammien käyttämät värit, jos oletus värit eivät ole sopivat. Lisäksi X-akselin asetuksista määritetään pylväsdiagrammin piirtämisen aikaväli. Esitetyissä kuvaajissa aikavälinä käytetään minuuttia. Aikaväli on muokattavissa millisekunnista vuoteen. Lopuksi jokainen kuvaaja tallennetaan.



KUVIO 24. CINR-kuvaaja

6.7.3 Kibana raportointinäkymä (Dashboard)

Kibanan Dashboard -välilehdellä muodostetaan aiemmin luoduista erillisistä kuvaajista raportointinäkymä. Raportointinäkymään voidaan sijoittaa useita kuvaajia haluttuihin paikkoihin. Kuvaajien sijainti ja koko voidaan määrittää käyttötarkoituksen mukaan. Lopuksi näkymä tallennetaan. Seuraavana on esitetty yhden yhteysvälin radioarvot-raportointinäkymä (KUVIO 25).



KUVIO 25. Radioarvot-raportointinäkymä

Pylväsdiagrammien keskivaiheella on havaittavissa selkeä radioyhteyden laadun heikkeneminen. Tällöin CINR ja RSSI:n huonontuessa myös modulaatio laskee. Radioyhteyden laadun heikkeneminen ei kuitenkaan näy kovin suuresti alimmassa tiedonsiirtonopeutta esittävässä kuvaajassa.

Radioarvot-näkymässä päästään selaamaan kuvaajia aikaan perustuen siten, että kaikki kuvaajat näyttävät arvot saman aikaleiman kohdalta. Kuvaajien yhden aikainen liikkuminen ajan suhteen helpottaa huomattavasti arvojen vertailua. Kuvaajia on mahdollista selata ajan(X-

akseli) suhteen niin pitkälle, kuin tallennettuja arvoja indeksikuviossa on olemassa.

6.8 Kibanan raportointinäkymän monistaminen

Jokaista tarkasteltavaa yhteysväliä varten täytyy Kibanaan luoda oma raportointinäkymä. Raportointinäkymän voi tehdä aina käyttöliittymällä, mutta se on huomattavasti työläämpää kuin alkuperäisen näkymän monistaminen ja muokkaaminen.

Raportointinäkymän monistaminen ei ole suoraan mahdollista vapaasti käytettävässä Kibanan versiossa. Monistaminen on kuitenkin kohtuullisen helposti toteutettavissa, kun haluttu raportointinäkymä kaikkine kuvaajineen viedään ensin tekstitiedostoon. Tekstitiedostossa muokataan kenttien nimet ja kuvaajien tiedot halutuiksi ja tallennetaan tiedosto uudella nimellä. Nämä toimenpiteet toistettiin kaikkien raportointinäkymien osalta. Tässä vaiheessa korostuu aikaisemmin Logstashissa määritettyjen uusien kenttien nimien rakenteen selkeys. Mitä selkeämmällä ja helpoimmin korvattavalla rakenteella kenttä on muodostettu, sitä helpompi ne on muodostaa tekstinkäsittelyohjelman ”etsi korvaa” -toiminnolla.

Maksullisessa Premium X-pack-versiossa raportointinäkymän kopiointi työkalut olisivat valmiina, mutta standardiversiossa niitä ei ole käytettävissä. Kyseessä ei kuitenkaan ole iso työmäärä, kun aikaisemmin Logstashin suodatinasetuksissa kaikkien yhteysvälien kentät on nimetty uudelleen tietyn logiikan mukaisesti.

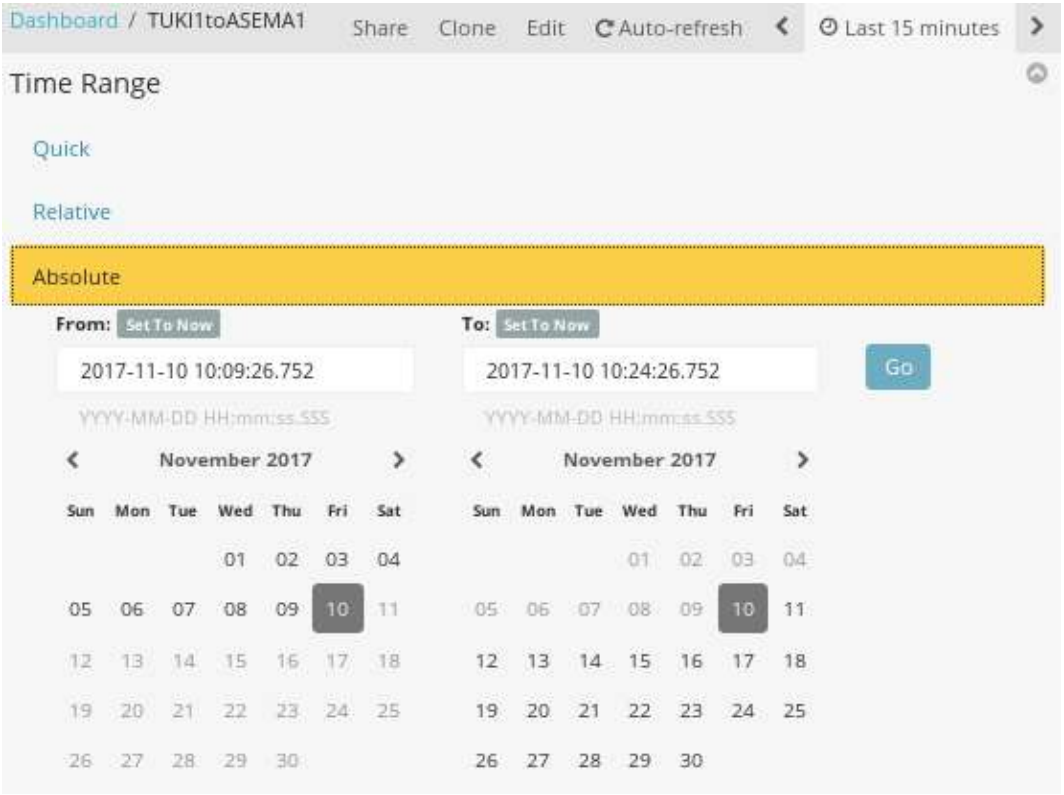
6.9 Kibana raportointinäkymän päivittäminen

Raportointinäkymän päivitysvalintoihin on useita eri tapoja. Näkymää voidaan päivittää automaattisesti tietyin väliajoin tai tarkasteluaikaväli voidaan määrittää jopa sekunnin tuhannesosan tarkkuudella. Näin ollen rajoitteeksi jää enää käytetyn lokiaineiston esiintymistiheys.

Automaattinen päivitys saadaan käyttöön valittaessa raportointinäkymän oikeasta yläkulmasta kellopainike ja tämän jälkeen avautuvaa Auto-refresh-painiketta (KUVIO 26). Päivityksen aikaväli vaihtoehtoja on useita. Alle minuutin päivitystahdilla päästään hyvin työn vaatimaan lähes reaaliaikaiseen näkymä vaatimukseen.

KUVIO 26. Raportointinäkymän automaattinen päivitys

Aikavälin määrittäminen manuaalisesti onnistuu useilla eri tavoilla. Aikaväli voidaan määrittää tarkasti raportointinäkymän kellon toiminnoilla tai haluttu aikaväli voidaan valita suoraan kuvaajasta. Näkymä sivun aikavalinnoissa on kolme vaihtoehtoa (KUVIO 27): nopea (Quick), suhteellinen (Relative) tai absoluuttinen (Absolute). Nopeassa valinnassa on valittavissa valmiita aikavälejä 15 minuutista viiteen vuoteen. Suhteellisessa valinnassa tarkkailuaikaväli voidaan sitoa joko alusta tai lopusta nykyhetkeen. Absoluuttisessa valinnassa tarkkailu välin alku ja loppu voidaan määrittää tarkasti käyttäen kalenterinäkymää.



KUVIO 27. Raportointinäkymän aikavalinta

7 TUTKIMUSTULOKSET

Tämän opinnäytetyön tarkoituksena oli selvittää, millä sovellusohjelmalla JSON-muotoisen lokitiedon sisältö saadaan esitettyä graafisesti. Tarkoituksena oli myös selvittää, kuinka tarvittava osa lokitiedosta saadaan esitettyä graafisesti lähes reaaliajassa. Lisäksi tässä opinnäytetyössä selvitettiin sitä, kuinka suodatus kannattaa toteuttaa, kun kyseessä on valtava määrä tietoa ja vain murto-osa tarvitaan käyttöön.

Työssä valituiksi sovellusohjelmiksi päätyivät Elasticin tarjoamat neljä sovellusta: Filebeat, Logstash, Elasticsearch ja Kibana. Jokaisella sovelluksella on oma tehtävänsä, joten kaikki neljä sovellusta vaaditaan työn toteutukseen. Filebeat lähettää lokitiedon lokia tuottavasta järjestelmästä Logstashille käsiteltäväksi. Logstash suodattaa ylimääräisen datan pois ja muokkaa jäljelle jäävän lokitiedon Kibanan vaatimaan esitettävään muotoon. Logstash tallentaa lokitiedon Elasticsearchiin, josta Kibana lukee sen Web-käyttöliittymään. Kibanassa on useita kuvaajia ja useita raportointinäkymiä halutun tiedon esittämiseen.

Graafisen esityksen reaaliaikavaatimus saadaan toteutettua, kun lokitietoa lähetetään, muokataan, tallennetaan ja esitetään koko ajan. Välissä ei saa olla yhtään käsin tehtävää toimenpidettä. Toteutuksessa kaikki sovellukset käsittelevät dataa sitä mukaa, kun sitä syntyy. Ainoana poikkeuksena tässä on Kibanan raportointinäkymä, joka ei automaattisesti tunnista, että uutta dataa on esitettävänä. Raportointinäkymä voidaan kuitenkin säätää päivittymään jopa viiden sekunnin välein. Näin ollen työn toteutuksen kannalta se riittää hyvin reaaliaikavaatimuksen toteutumiseen.

Reaaliaikavaatimukseen liittyy myös olennaisesti aikaleiman käyttö. Tämä on huomioitava kaikissa sovelluksissa, joiden kautta lokitieto kulkee. Näin varmistutaan myös siitä, että käyttäjä pääsee tarkistamaan, millä hetkellä lokitieto on syntynyt ja vertaaminen nykyhetkeen mahdollistuu.

Sovellusohjelman valinnan jälkeen yksi keskeisimmistä ratkaistavista asioista opinnäyteyön toteutuksessa oli löytää sopiva tapa suodattaa ylimääräinen lokitieto pois. Lokitiedon suodatukseen työssä löydettiin tehokas tapa. Suodatus toteutetaan siten, että tarvittava osa lokitiedoista kopioidaan muokkauksen yhteydessä uuteen tietorakenteeseen. Samalla tehdään lokitiedon siivomainen ja muokkaaminen esityksen vaatimaan muotoon. Tarpeellisen lokitiedon kopioinnin ja muokkauksen jälkeen koko alkuperäinen lokitieto poistetaan. Tällaisella tekniikalla vältytään suurelta lokitiedon poistoluettelolta ja saadaan poistettua myös kaikki tunnistamaton lokitieto. Tämän työn toteutuksen kannalta kaikki tunnistamaton lokitieto on tarpeetonta.

Työssä hyödynnetään noin 400 kentän lokitiedot. Alkuperäisessä lokitiedossa kenttiä on useita tuhansia. Näin ollen kaikkien poistettavien kenttien luetteleminen ei olisi ollut järkevää. Se ei olisi myöskään mahdollista, koska kenttien nimet rakentuvat verkonlaitteiden tietojen perusteella ja etukäteen ei voida tietää, mitä laitteita verkkoon voi jossakin vaiheessa liittyä. Alkuperäinen lokitieto rakentuu vain muutamaan oksaan, joiden poistaminen juuresta on yksinkertaista.

Tämä suodatusperiaate toimii vain tässä työssä, kun etukäteen tiedetään, mitä tietoa tarvitaan käyttöön ja millaisessa tietorakenteessa se sijaitsee. Mikäli näitä tietoja ei olisi, suodattimien toiminta jouduttaisiin suunnittelemaan kokonaan toisella periaatteella. Valitulla suodatusratkaisulla ei myöskään voida ottaa käyttöön mitään uutta, entuudestaan tuntematonta lokitietoa ilman muutoksia.

8 JOHTOPÄÄTÖKSET JA JATKOKEHITYS

Teoria- ja käytännön osuuden toteutus on antanut opinnäytetyön tekijälle hyvän kuvan siitä, millaisia mahdollisuuksia sovellukset antavat lokitiedon käsittelyyn ja esittämiseen. Työn myötä opinnäytetyön tekijälle on kehittynyt valmiudet tehdä muihinkin lokitiedostoihin liittyviä visualisointeja kohtuullisen nopeasti. Elastic-sovelluspaketin käyttö on kuitenkin kohtuullisen yksinkertaista, kun sovelluksiin on päässyt sisälle.

Työn käytännön osuus on toteutettu neljällä Elasticin sovelluksella: Filebeat, Logstash, Elasticsearch ja Kibana. Nämä sovellukset mahdollistavat hyvin erityyppisen datan vastaanottamisen, muokkaamisen ja esittämisen. Elastic sovellusten käyttö ei rajoitu ainoastaan JSON-muotoisen datan käsittelyyn, vaan sovellukseen saatavat liitännäiset mahdollistavat laajan käyttöalueen. Lisäksi valitusta grafiikka sovelluksesta Kibanasta löytyy paljon erilaisia esitysmahdollisuuksia.

Käytetyn JSON-lokitiedoston laajuus yllätti työn edetessä. Sopivilla työkaluilla ja menetelmillä tarvittavat lokitiedon osat saatiin kuitenkin otettua käyttöön, vaikka käytettyjen sovellusten rajat tulivatkin välillä vastaan.

Työssä käytetty lokitiedon suodatusmalli on tehokas tilanteeseen, jossa hyödynnettävää tietoa on vain murto-osa koko lokitiedosta. Työssä käytetyssä suodatusmallissa tarvittava lokitieto kopioidaan uuteen rakenteeseen ja alkuperäinen tieto poistetaan. Mallin käyttö edellyttää kaiken tarvittavan lokitiedon rakenteen tuntemista ja sen rakentumista puumaisesti. Puumaisessa rakenteessa kaikki lokitieto voidaan poistaa suoraan juuresta.

Elastic-sovelluksiin julkaistaan useita päivityksiä vuodessa. Sovelluspäivitysten myötä julkaistaan uusia ominaisuuksia ja korjataan havaittuja virheitä. Käyttäjän valitsemasta lisenssistä riippuen sovelluksille on myös tarjolla valmistajan tuki. Sovelluspäivitysten saaminen ei vaadi maksullisen lisenssin hankkimista. Sovellukset skaalautuvat yhden

kannettavan tietokoneen ympäristöstä klusteroituihin palvelinratkaisuihin. Näin ollen ympäristön laajentaminen käyttötarkoituksen ja tarpeen mukaan on varsin yksinkertaista. Nämä Elasticin tarjoamat mahdollisuudet mahdollistavat hyvin myös koulutus- ja opetuskäytön.

Työn tekeminen on antanut hyvän kuvan lokien visualisoinnista ja siitä mihin Elasticin ohjelmilla pystytään. Entuudestaan yksikään sovellus ei ollut tekijälle tuttu. Sovellusten käytön aloitus ilman koulutusta on kohtuullisen vaativaa, koska ei ole olemassa mitään yksittäistä ohjetta, jonka avulla työssä käytetty iso JSON-tiedosto pystytäisiin visualisoimaan. Työn tekeminen vaati hyvää perehtymistä valmistajan dokumentaatioon ja käytännön esimerkkien etsimistä eri keskustelupalstoilta. Parhaiten tietoa löytyi github.com sivuston eri keskusteluryhmistä.

Opinnäytetyön lopputuloksena lokitiedon sisällöstä pystytään esittämään vaadittavat osat reaaliaikaisina kuvaajina. Kuvaajissa esitettäviä tietoja ovat vaadittujen yhteysvälien osalta tiedonsiirtonopeus, CINR-, RSSI- ja modulaatio-arvot. Lisäksi kaikkiin tutkimuskysymyksiin on löydetty vastaukset. Nyt nähdään selkeämmin, miten modulaatio käyttäytyy suhteessa radioarvoihin, kun tiedot ovat samassa näkymässä ajan funktiona. Näin ollen voidaan todeta, että alussa työlle asetettuihin tavoitteisiin on päästy.

Ennen mittausjärjestelmän varsinaista käyttöönottoa on tehtävä testikäyttö. Vaatii vielä huomattavan paljon työtä, että voidaan varmistua mittausjärjestelmän oikeasta toiminnasta. Yksi testikäytön tärkeimmistä kohteista on kaikkien tietokenttien oikeellisuuden tarkastaminen verrattuna alkuperäiseen tietoon ja aikaleimaan.

Testikäytön yhteydessä on syytä kiinnittää huomioita myös käyttöliittymään. Käyttöliittymään tulee varmasti muutoksia ja parannuksia testikäytön havaintojen ja kokemusten perusteella. Käyttöliittymän kehityskohteita voivat olla esimerkiksi kenttien järjestyksen vaihtaminen sopivammaksi tietojen vertailuun. Lisäksi koekäytössä saatetaan havaita tarve saada vielä lisää tietoa näkyviin käyttöliittymään.

Mittausjärjestelmän toteutuksen hyötynä on luonnollisesti se, että ylläpitokyky pystytään säilyttämään omassa hallinnassa. Ylläpitokyvyn lisäksi on myös kyky kehittää nykyistä mittausjärjestelmää sekä luoda uusia mittausjärjestelmiä uusia sovelluksia tai laitteita varten. Ulkopuolisia yrityksiä tai konsultteja ei välttämättä tarvita muutostöiden ja kehitystarpeiden toteuttamiseen.

LÄHTEET

Courcera. 2017. Big data management. [viitattu 19.11.2017] Saatavissa: https://www.coursera.org/learn/big-data-management/lecture/irelt/summary-of-introduction-to-big-data-part-2

CSEstack. 2017. DIKW-pyramiidi [viitattu 29.11.2017] Saatavissa: http://www.csestack.org/dikw-pyramid-model-difference-between-data-information/

Dietrich D., Heller B. & Yang B. 2015. Data Science and Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data. Indianapolis: Dietrich Willey & Sons, Inc

Elastic. 2017. Elastic Products. [viitattu 5.5.2017] Saatavissa: https://www.elastic.co/products/elasticsearch

Elasticsearch Docs. 2017a. Elasticsearch Reference [5.6] » Getting Started. [viitattu 4.10.2017] Saatavissa: https://www.elastic.co/guide/en/elasticsearch/reference/5.6/getting-started.html

Elasticsearch Docs. 2017b. Elasticsearch Reference [5.6] » Getting Started » Basic Concepts. [viitattu 4.10.2017] Saatavissa: https://www.elastic.co/guide/en/elasticsearch/reference/5.6/_basic_concepts.html

Elasticsearch Docs. 2017c. Elasticsearch Reference [5.6] » Indices APIs » Index Templates. [viitattu 5.10.2017] Saatavissa: https://www.elastic.co/guide/en/elasticsearch/reference/5.6/indices-templates.html

Filebeat. 2017. Filebeat Products. [viitattu 10.8.2017] Saatavissa: https://www.elastic.co/products/beats/filebeat

Filebeat Docs. 2017a. Filebeat Reference [5.6] » Getting Started With Filebeat. [viitattu 10.8.2017] Saatavissa:

https://www.elastic.co/guide/en/beats/filebeat/5.6/filebeat-getting-started.html

Filebeat Docs. 2017b. Filebeat Reference [5.6] » » Migrating from Logstash Forwarder to Filebeat » Updating the Registry File. [viitattu 10.8.2017] Saatavissa: https://www.elastic.co/guide/en/beats/filebeat/5.6/migration-registry-file.html

Filebeat Docs. 2017c. Filebeat Reference [5.6] » Configuring Filebeat » Processors. [viitattu 10.8.2017] Saatavissa: https://www.elastic.co/guide/en/beats/filebeat/5.6/configuration-processors.html

Filebeat Docs. 2017d. Filebeat Reference [5.6] » Configuring Filebeat » Managing Multiline Messages. [viitattu 10.8.2017] Saatavissa: https://www.elastic.co/guide/en/beats/filebeat/5.6/multiline-examples.html

Frank B. 2012. Taming the big data tidal wave: finding opportunities in huge data streams with advanced analytics. Hoboken: John Willey & Sons, Inc

GDPR haltuun: Näin valmistaudut EU:n uuden tietosuoja-asetuksen käyttöönottoon. Suomen Asiakastieto Oy. [viitattu 17.2.2018] Saatavissa: https://www.asiakastieto.fi/web/fi/gdpr.html

Gormley C. & Tong Z. 2015a. Elasticsearch: The Definitive Guide [2.x] » Getting Started. Sebastopol: O`Reilly Media. [viitattu 4.10.2017] Saatavissa: https://www.elastic.co/guide/en/elasticsearch/guide/2.x/index.html

Gormley C. & Tong Z. 2015b. Elasticsearch: The Definitive Guide [2.x] » Getting Started » Life Inside a Cluster. Sebastopol: O`Reilly Media. [viitattu 4.10.2017] Saatavissa: https://www.elastic.co/guide/en/elasticsearch/guide/2.x/distributed-cluster.html

JSON. 2017. Introducing JSON [viitattu 5.5.2017] Saatavissa: http://www.json.org/

Järvinen P. & Järvinen A. 2011. Tutkimustyön metodeista. Tampere: Opinpajan kirja.

Kanerva J. 2017. Parempia esityksiä. Opas presentaation tekijöille tiedon ja data esittämiseen. PDF-julkaisu. [viitattu 1.11.2017] Saatavissa: https://infograafikko.fi

Kibana. 2017. Kibana products. [viitattu 5.5.2017] Saatavissa: https://www.elastic.co/products/kibana

Kibana Docs. 2017a. Kibana User Guide [5.6] » Introduction. [viitattu 5.10.2017] Saatavissa: https://www.elastic.co/guide/en/kibana/5.6/introduction.html

Kibana Docs. 2017b. Kibana User Guide [5.6] » Getting Started. [viitattu 5.10.2017] Saatavissa: https://www.elastic.co/guide/en/kibana/5.6/getting-started.html

Kibana Docs. 2017c. Kibana User Guide [5.6] » Discover. [viitattu 5.10.2017] Saatavissa: https://www.elastic.co/guide/en/kibana/5.6/discover.html

Kibana Docs. 2017d. Kibana User Guide [5.6] » Visualize. [viitattu 5.10.2017] Saatavissa: https://www.elastic.co/guide/en/kibana/5.6/visualize.html

Kibana Docs. 2017e. Kibana User Guide [5.6] » Dashboard. [viitattu 5.10.2017] Saatavissa: https://www.elastic.co/guide/en/kibana/5.6/dashboard.html

Kibana Docs. 2017f. Kibana User Guide [5.6] » Timelion. [viitattu 5.10.2017] Saatavissa: https://www.elastic.co/guide/en/kibana/5.6/timelion.html

Kibana Docs. 2017g. Kibana User Guide [5.6] » Dev Tools. [viitattu 5.10.2017] Saatavissa: https://www.elastic.co/guide/en/kibana/5.6/devtools-kibana.html

Kibana Docs. 2017h. Kibana User Guide [5.6] » Management. [viitattu 5.10.2017] Saatavissa: https://www.elastic.co/guide/en/kibana/5.6/management.html

Koponen J., Hildén J. & Vapaasalo T. 2016. Tieto näkyväksi. Informaatio muotoilun perusteet. Helsinki: Aalto ARTS Books.

Kosola J. 2013. Digitaalinen taistelukenttä: informaatioajan sotakoneen tekniikka. 3. Painos. Helsinki: Maanpuolustuskorkeakoulu, Sotatekniikan Laitos

Kyberturvallisuus. 2016. Viestintävirasto. [Viitattu 29.9.2017] Saatavissa: https://www.viestintavirasto.fi/kyberturvallisuus/tietoturvanyt/2016/03/ttn20160309174 2.html

Lehto A. & Räisänen A. 2007. Radiotekniikan perusteet. Helsinki: Otatieto

Logstash. 2017. Logstash Introduction. [viitattu 5.5.2017] Saatavissa: https://www.elastic.co/products/logstash

Logstash Docs. 2017a. Logstash Reference [5.6] » Getting Started with Logstash » Stashing Your First Event. [viitattu 5.5.2017] Saatavissa: https://www.elastic.co/guide/en/logstash/5.6/first-event.html

Logstash Docs. 2017b. Logstash Reference [5.6] » Setting Up and Running Logstash » Logstash Directory Layout. [viitattu 2.10.2017] Saatavissa: https://www.elastic.co/guide/en/logstash/5.6/dir-layout.html

Logstash Docs. 2017c. Logstash Reference [5.6] » Setting Up and Running Logstash » Settings File. [viitattu 2.10.2017] Saatavissa: https://www.elastic.co/guide/en/logstash/5.6/logstash-settings-file.html

Logstash Docs. 2017d. Logstash Reference [5.6] » Setting Up and Running Logstash » Logstash Configuration Files. [viitattu 2.10.2017]

Saatavissa: https://www.elastic.co/guide/en/logstash/5.6/config-setting-files.html

Logstash Docs. 2017e. Logstash Reference [5.6] » How Logstash Works. [viitattu 2.10.2017] Saatavissa: https://www.elastic.co/guide/en/logstash/5.6/pipeline.html

Logstash Docs. 2017f. Logstash Reference [5.6] » Filter plugins » Json filter plugin. [viitattu 2.10.2017] Saatavissa: https://www.elastic.co/guide/en/logstash/5.6/plugins-filters-json.html

Logstash Docs. 2017g. Logstash Reference [5.6] » Codec plugins. [viitattu 3.10.2017] Saatavissa: https://www.elastic.co/guide/en/logstash/5.6/codec-plugins.html

Logstash Docs. 2017h. Logstash Reference [5.6] » Working with plugins. [viitattu 3.10.2017] Saatavissa: https://www.elastic.co/guide/en/logstash/5.6/working-with-plugins.html

Lokiohje. 2009. Valtionvarainministeriön verkkosivut. Saatavissa: https://www.vahtiohje.fi/web/guest/3/2009-lokiohje

Nuaymi L. 2007. WiMAX: Technology for Broadband Wireless Access. West Sussex: John Wiley & Sons, Ltd

Salo I. 2013. Big data: Tiedon vallankumous. Helsinki: Docendo

Telea A. 2015. Data Visualization: Principles and Practice. Boca raton: CRC Press

Viestintävirasto. 2017. Taajuusjakotaulukko. [viitattu 31.10.2017] Saatavissa: https://www.viestintavirasto.fi/taajuudet/radiotaajuuksienkaytto/taajuusjakotaulukko.html

Wiki freepascal. 2017 [viitattu 4.5.2017] Saatavissa: http://wiki.freepascal.org/JSON/fi

X-Pack Docs. 2017a. X-Pack for the Elastic Stack [5.6] » Introduction. [viitattu 6.11.2017] Saatavissa: https://www.elastic.co/guide/en/x-pack/5.6/xpack-introduction.html

X-Pack Docs. 2017b. X-Pack for the Elastic Stack [5.6] » Reporting from Kibana. [viitattu 6.11.2017] Saatavissa: https://www.elastic.co/guide/en/x-pack/5.6/xpack-reporting.html

X-Pack Docs. 2017c. Kibana User Guide [5.6] » Graphing Connections in Your Data. [viitattu 6.11.2017] Saatavissa: https://www.elastic.co/guide/en/kibana/5.6/xpack-graph.html

X-Pack Docs. 2017d. X-Pack for the Elastic Stack [5.6] » License Management. [viitattu 6.11.2017] Saatavissa: https://www.elastic.co/guide/en/x-pack/5.6/license-management.html

X-Pack Docs. 2017e. X-Pack for the Elastic Stack [5.6] » Securing Elasticsearch and Kibana. [viitattu 6.11.2017] Saatavissa: https://www.elastic.co/guide/en/x-pack/current/xpack-security.html

X-Pack Docs. 2017f. X-Pack for the Elastic Stack [5.6] » Monitoring the Elastic Stack. [viitattu 6.11.2017] Saatavissa: https://www.elastic.co/guide/en/x-pack/5.6/xpack-monitoring.html

X-Pack Docs. 2017g. X-Pack for the Elastic Stack [5.6] » Alerting on Cluster and Index Events. [viitattu 6.11.2017] Saatavissa: https://www.elastic.co/guide/en/x-pack/5.6/xpack-alerting.html

X-Pack Docs. 2017h. X-Pack for the Elastic Stack [5.6] » Machine Learning in the Elastic Stack. [viitattu 6.11.2017] Saatavissa: https://www.elastic.co/guide/en/x-pack/5.6/xpack-ml.html

Yuk M. & Diamond S. 2014. Data Visualization For Dummies. Hoboken: John Willey & Sons, Inc